Adaptive visual sampling

Raja, Yogesh

Adaptive Visual Sampling

Yogesh Raja

Submitted to the University of London in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Queen Mary, University of London

2010

Adaptive Visual Sampling

Yogesh Raja

Abstract

Various visual tasks may be analysed in the context of *sampling* from the visual field. In visual psychophysics, human visual sampling strategies have often been shown at a high-level to be driven by various information and resource related factors such as the limited capacity of the human cognitive system, the quality of information gathered, its relevance in context and the associated efficiency of recovering it. At a lower-level, we interpret many computer vision tasks to be rooted in similar notions of contextually-relevant, dynamic sampling strategies which are geared towards the *filtering* of pixel samples to perform reliable object association. In the context of object tracking, the reliability of such endeavours is fundamentally rooted in the continuing relevance of object models used for such filtering, a requirement complicated by real-world conditions such as dynamic lighting that inconveniently and frequently cause their rapid obsolescence. In the context of recognition, performance can be hindered by the lack of learned context-dependent strategies that satisfactorily filter out samples that are irrelevant or blunt the potency of models used for discrimination. In this thesis we interpret the problems of visual tracking and recognition in terms of dynamic spatial and featural sampling strategies and, in this vein, present three frameworks that build on previous methods to provide a more flexible and effective approach.

Firstly, we propose an adaptive spatial sampling strategy framework to maintain statistical object models for real-time robust tracking under changing lighting conditions. We employ colour features in experiments to demonstrate its effectiveness. The framework consists of five parts: (a) Gaussian mixture models for semi-parametric modelling of the colour distributions of multi-colour objects; (b) a constructive algorithm that uses cross-validation for automatically determining the number of components for a Gaussian mixture given a sample set of object colours; (c) a sampling strategy for performing fast tracking using colour models; (d) a Bayesian formulation enabling models of object and the environment to be employed together in filtering samples by discrimination; and (e) a selectively-adaptive mechanism to enable colour models to cope with changing conditions and permit more robust tracking.

Secondly, we extend the concept to an adaptive spatial and featural sampling strategy to deal with very difficult conditions such as small target objects in cluttered environments undergoing severe lighting fluctuations and extreme occlusions. This builds on previous work on dynamic feature selection during tracking by reducing redundancy in features selected at each stage as well as more naturally balancing short-term and long-term evidence, the latter to facilitate model rigidity under sharp, temporary changes such as occlusion whilst permitting model flexibility under slower, long-term changes such as varying lighting conditions. This framework consists of two parts: (a) Attribute-based Feature Ranking (AFR) which combines two attribute measures; discriminability and independence to other features; and (b) Multiple Selectively-adaptive Feature Models (MSFM) which involves maintaining a dynamic feature reference of target object appearance. We call this framework Adaptive Multi-feature Association (AMA).

Finally, we present an adaptive spatial and featural sampling strategy that extends established Local Binary Pattern (LBP) methods and overcomes many severe limitations of the traditional approach such as limited spatial support, restricted sample sets and ad hoc joint and disjoint statistical distributions that may fail to capture important structure. Our framework enables more compact, descriptive LBP type models to be constructed which may be employed in conjunction with many existing LBP techniques to improve their performance without modification. The framework consists of two parts: (a) a new LBP-type model known as Multiscale Selected Local Binary Features (MSLBF); and (b) a novel binary feature selection algorithm called Binary Histogram Intersection Minimisation (BHIM) which is shown to be more powerful than established methods used for binary feature selection such as Conditional Mutual Information Maximisation (CMIM) and AdaBoost.

Submitted to the University of London in partial fulfilment of the requirements for

the degree of Doctor of Philosophy

Queen Mary, University of London

2010

Declaration

I hereby declare that this thesis has been composed by myself and that it describes my own work. It has not been submitted, either in the same or different form, to this or any other university for a degree. All verbatim extracts are distinguished by quotation marks, and all sources of information have been acknowledged.

Some parts of the work have previously been published as:

1. S.J. McKenna, S. Gong and Y.Raja. Face Recognition in Dynamic Scenes. In *British Machine Vision Conference*, pages 140-151, Essex, U.K. 1997.

2. Y. Raja, S.J. McKenna and S. Gong. Segmentation and Tracking using Colour Mixture Models. In *Asian Conference on Computer Vision*, pages 607-614, Hong Kong 1998.

3. S.J.McKenna, Y. Raja and S. Gong. Object Tracking using Adaptive Colour Mixture Models. In *Asian Conference on Computer Vision*, pages 615-622, Hong Kong 1998.

4. Y. Raja, S.J. McKenna and S. Gong. Tracking and Segmenting People in Varying Lighting Conditions using Colour. In *International Conference on Automatic Face and Gesture Recognition*, pages 228-233, Nara, Japan 1998.

5. Y. Raja, S.J. McKenna and S. Gong. Colour Model Selection and Adaptation in Dynamic Scenes. In *European Conference on Computer Vision*, pages 460-474, Freiburg, Germany 1998.

6. Y. Raja, S.J. McKenna and S. Gong. Using Colour for Robust Object Tracking and Segmentation. In *Noblesse Workshop on Non-linear Model Based Image Analysis*, pages 199-204, Glasgow, Scotland 1998.

7. S.J. McKenna, S. Gong and Y.Raja. Modelling Facial Colour and Identity with Gaussian Mixtures. In *Pattern Recognition*, 31(12):1883-1892, 1998.

8. S.J.McKenna, Y. Raja and S. Gong. Tracking Colour Objects using Adaptive Mixture Models. In *Image and Vision Computing*, 17(3):225-231, 1999.

9. Y. Raja and S. Gong. Sparse Multiscale Local Binary Patterns. In *British Machine Vision Conference*, pages 799-808, Edinburgh, U.K. 2006.

10. Y. Raja and S. Gong. Robust Tracking by Adaptive Multi-feature Association. Submitted to *International Journal of Computer Vision*, 2010.

Yogesh Raja

London, January 2010.

Acknowledgements

Firstly, I would like to thank my supervisor Professor Shaogang Gong for his perpetual patience, encouragement and invaluable guidance. I doubt I could have had a more capable or accommodating advisor and he has my deepest gratitude.

I would also like to thank the various academic staff, students and associates with whom I have worked and socialised over the years, in particular Stephen McKenna, Yakup Paker, Eng-Jon Ong, Richard Howarth, Dennis Parkinson, Jamie Sherrah, Hayley Hung, Tao Xiang, Lourdes Agapito, Lukasz Zalewski, David Russell, Jian Li, Caifeng Shan, Samuel Pachoud, Chen Change Loy, Bryan Prosser, Bob Koger and Milan Verma.

I am grateful to all the friendly and highly competent administrative and systems support staff in the department for enabling things to run smoothly and efficiently.

Finally, I am indebted to my family and friends and in particular my mother Madhuvanti for her unlimited patience, support and understanding, my sister Nimisha for her encouragement and my niece Freya for being a perpetual ray of sunshine. I dedicate this work to them.

Contents

1 Introduction **35**

1.1 Visual sampling in humans 35

1.2 Visual sampling in computer vision 39

1.3 Object association 41

1.4 Approach 44

1.4.1 Colour feature sampling for tracking 45

1.4.2 Multi-feature sampling for tracking 47

1.4.3 Local Binary Pattern feature sampling for recognition 48

1.5 Contributions 49

1.6 Thesis outline 51

2 Literature Review **52**

2.1 Feature selection for classification 53

2.1.1 Ranking features 55

2.1.2 Feature subset selection 57

2.1.3 Filters vs wrappers 60

2.2 Sampling for tracking 63

2.2.1 Modelling foreground 65

2.2.2 Modelling background 70

2.2.3 Foreground-background classification 75

2.3 Sampling for recognition 83

2.3.1 Local Binary Patterns 85

2.3.2 Further developments of LBP 88

2.3.3 LBP-based recognition 91

2.4 Summary 95

3 Colour Feature Sampling for Tracking **98**

3.1 Scope of the problem . . . 99

3.1.1 Modelling colour distributions . . . 100

3.1.2 The model order selection problem . . . 100

3.1.3 Tracking with colour cues . . . 101

3.1.4 Dealing with changing lighting . . . 102

3.2 Gaussian mixture models of colour . . . 102

3.3 Automatic model order selection . . . 103

3.3.1 Splitting components . . . 105

3.3.2 A constructive algorithm for model-order selection . . . 106

3.4 Fast colour-based tracking . . . 106

3.5 Modelling colour in environmental context . . . 109

3.6 Coping with change . . . 110

3.6.1 On-line model adaptation . . . 111

3.6.2 Selective adaptation . . . 112

3.7 Experiments . . . 113

3.7.1 Fast colour-based tracking . . . 113

3.7.2 Modelling colour in environmental context . . . 114

3.7.3 Coping with change . . . 114

3.8 Discussion . . . 117

3.9 Summary . . . 119

4 Multi-Feature Sampling for Tracking **121**

4.1 Scope of the problem . . . 122

4.2 Adaptive Multi-feature Association . . . 125

4.2.1 Attribute-based Feature Ranking . . . 126

4.2.2 Multiple Selectively-adaptive Feature Models . . . 128

4.2.3 Tracking by Adaptive Multi-feature Association . . . 130

4.3 Experiments . . . 133

4.3.1 Datasets and Settings . . . 133

4.3.2 Implementations . . . 135

4.3.3 Scenario A: Feature selection performance . . . 136

4.3.4 Scenario B: Tracking under severe illumination changes . . . 137
4.3.5 Scenario C: Tracking under occlusion . . . 149
4.3.6 Scenario D: Tracking under severe occlusion and illumination changes . 156
4.4 Discussion . . . 166
4.5 Summary . . . 169

5 Local Binary Pattern Feature Sampling for Recognition 171
5.1 Scope of the problem . . . 172
5.2 Multiscale Selected Local Binary Features . . . 176
5.2.1 Texton selection by Binary Histogram Intersection Minimisation . . . 177
5.2.2 MSLBF classification procedure . . . 183
5.3 Experiments . . . 184
5.3.1 Texture recognition . . . 185
5.3.2 Face recognition . . . 188
5.3.3 Comparison of feature selection methods . . . 190
5.4 Discussion . . . 193
5.5 Summary . . . 195

6 Conclusions 201
6.1 Colour feature sampling for tracking . . . 201
6.1.1 Future work . . . 203
6.2 Multi-feature sampling for tracking . . . 203
6.2.1 Future work . . . 205
6.3 Local Binary Pattern feature sampling for recognition . . . 206
6.3.1 Future work . . . 207

A On-line Gaussian Mixture Adaptation 208

Bibliography 210

List of Figures

1.1 Example of automated segmentation and classification of malarial parasites in a blood specimen. True positives (TP), true negatives (TN) and false negatives (FN) are highlighted. (Tek et al. [217]). 36

1.2 Detection of road boundaries and road junctions for automated vehicle navigation. Detected roadside boundaries (the thick black and white lines) are used to estimate the trajectory of the road and the position/orientation of junctions (the thin black lines). (Ekinci et al. [50]). 37

1.3 Detection of anomalous road crossing ("jaywalking") in automated public CCTV analysis (Hospedales et al. [85]). 38

1.4 Spatially regular pixel sampling of an image to reduce the required storage while maintaining the fundamental physical forms within. The right image is the same as the left but subsampled to contain only 1% of the pixel data. While there is less detail, the overall spatial shapes and relationships across the entire image are maintained. 40

1.5 Selective pixel sampling based on object membership. The right image retains only those pixels which are classified as belonging to the red portion of the rose in the left image. In this case, the sampling strategy is to accept or reject pixels based on the probabilistic likelihood of having been generated by a model representing either the red part of the flower or the rest of the image. 40

1.6 An example of a scenario for tracking a person (highlighted with a yellow arrow) through a crowd of people in a low-resolution, cluttered scene. Initialised in (a), the person moves across similarly coloured distractors (b), undergoes severe occlusion (c) and experiences chromatic change due to lighting fluctuations in (d). Such conditions put an insurmountable strain on features to stay representative of a target object during tracking to the exclusion of proximal distractors, suggesting a more appropriate strategy of dynamically selecting the most discriminative feature type at any given time as well as updating their corresponding models. . . 43

1.7 Visual tasks such as tracking and recognition (in both humans and machines) involve the visual sampling of input stimuli given a specific sampling strategy. Many computer vision sampling strategies are static both in terms of features and models used. The focus of this work is on developing flexible computer vision sampling strategies (dashed box) for tracking and object recognition which consist of feature selection and/or model adaptation to cater for the needs of contextual relevance and changing object appearance over time. 45

1.8 An example Spatial-Featural Volume for an image. Each "slice" constitutes a property or transformation of the original image, such as hue or edge-orientation at each pixel. Slices may themselves comprise multiple layers, such as the Histogram-of-Gradients HoG [41] descriptor comprising the statistics of oriented gradients in the fixed neighbourhood of each pixel. Vision tasks generally involve the evaluation of samples with the $2D$ x-y spatial domain but feature selection may also be viewed in this context as sampling from the third, "featural" dimension. 46

2.1 The "filter" approach to feature subset selection (Kohavi and John [102]). The feature selection algorithm is self-contained and independent of the induction algorithm to be applied. 54

2.2 The "wrapper" approach to feature subset selection (Kohavi and John [102]). The selection algorithm is "wrapped" around the induction algorithm (classifier) which is used to evaluate the results of adding or subtracting features from the chosen subset. Because of this approach, wrapper methods tend to cater for the biases of individual induction algorithms in choosing the best features; conversely, it also results in a tendency to overfit when training datasets are small. . . 55

2.3 The computational intractability of exhaustive feature subset selection (Blum and Langley [15]). Selecting subsets from a pool of four features requires first choosing the first feature (left) and then subsequently adding features to complement the one(s) already chosen (moving rightwards). As the pool increases, the set of possible combinations at each stage become prohibitively large. Consequently, in practice most algorithms can only traverse suboptimal subpaths through the tree which are highly dependent on the initial starting point and the criteria used to evaluate features. 59

2.4 The Pfinder system developed by Wren et al. [234]. The left image is the input image. The middle image depicts the grouped chromatically coherent regions for blob model generation. The right image illustrates the resulting blob model, with each blob relating to a single coherent region as generated by the segmentation step. 72

2.5 An example of tracking using adaptive Gaussian mixtures to dynamically model per-pixel background colour (Stauffer and Grimson [206]). During tracking, pixels sufficiently deviated from their background models are considered to be foreground. The images show (from left to right); first the input image, followed by the image constructed from the means of the most probable Gaussians for each pixel. Third, a binary background/foreground image and finally the result of tracking on the foreground binary image. 72

2.6 Example of foreground extraction with an adaptive background model (Cavallaro and Ebrahimi [22]). The left column shows input images and the right column corresponding extracted regions. 73

2.7 Example of foreground extraction using spatial, spectral and temporal features in a unified Bayesian framework (Li et al. [111]). The scene contains difficult lighting and dynamic scene components (an escalator). 74

2.8 Centre-surround bounding box. The large red box denotes the region of interest and pixels are sampled from within. Background pixels correspond to all but the pixels within the smaller, central rectangle which denotes the foreground sample region. These samples are used for ranking features on-the-fly, as in Collins et al. [29]. 77

2.9 Features used for tracking an object must be adapted as the appearance of the object and background changes (Collins and Liu [30]). The source imagery (left column) is a low contrast aerial video of a car on a road. The car travels between sunny patches (top row) and shadow (bottom row). The best feature for tracking the car in sunlight (R-G) performs poorly in shadow. Similarly, the best feature for tracking through shadow (2G-B) does not perform as well in sunlight. 78

2.10 Sample video frames with ranked weight images (Collins and Liu [30]). Left column: frame with labelled object (inner box) and background pixels (outer box) pixels. Second-fourth columns: weight images corresponding to the features with highest, median and lowest variance ratio scores, respectively. The features with the higher variance ratio scores show the target object as more distinctive against the background. 79

2.11 Overview of tracking system with on-line, adaptive feature selection (Collins and Liu [30]). Samples of object and background pixels in the previous frame guide evaluation of candidate features, leading to a rank ordering of features based on discriminative ability. The top N best features are applied to the current frame to compute N weight images. A mean-shift process is applied to each weight image to compute a 2D location estimate. These N estimates are combined to determine the best location of the object in the current frame and the procedure iterates. . . 80

2.12 Illustration of ensemble tracking update (Avidan [4]). At each frame, the current ensemble is used to classify pixels taken from the foreground (centre) region and pixels from the background (surround) region as shown in the leftmost image. A confidence map for each pixel of the region of interest is computed (centre image) which is applied to mean-shift. The resulting fitted bounding box is used to train a new weak classifier (dashed line, rightmost image) which is integrated into the current ensemble. 81

2.13 Feature selection using an on-line version of AdaBoost (Grabner et al. [71]) . . . 81

2.14 Local binary pattern (Ojala et al. [154]). For each pixel, intensity values in the neighbourhood are thresholded by the centre value and transformed into a binary string according to the resulting sign. The decimal equivalent of the string is used as the feature value for that pixel. 86

2.15 Circular local binary pattern (Ojala et al. [155]). For each pixel, intensity values are sampled at subpixel level at a fixed distance and regular angular points surrounding the centre. Similar to the approach from Figure 2.14, they are then thresholded by the centre value and transformed into a binary string according to the sign of the result. The decimal equivalent of the string is used as the feature value for that pixel. 87

2.16 The Multi-scale Block Local Binary Pattern (MB-LBP) (Liao et al. [118]). Blocks of pixels are averaged over before being treated as a single value for LBP computation (here, each value comprises the average of a local 3x3 neighbourhood). This amounts to a subsampling of the image and enables more robustness against local noise as well as effectively increasing the spatial support area of derived patterns. 89

2.17 LBP patterns selected by a boosting procedure for facial expression recognition (Shan et al. [193], Shan and Gritti [196]). Each rectangle denotes a facial region corresponding to a selected weak LBP classifier trained on that region. They can be viewed as corresponding to key physical locations useful for discriminating between expressions. 90

2.18 The procedure for generating a Local Gabor Binary Pattern Histogram Sequence (LGBPHS) (Zhang et al. [249]). Normalised face images are Gabor filtered at multiple scales and orientations to derive Gabor Magnitude Pictures (GMPs) which are then processed with LBP operators to generate Local Gabor Binary Pattern (LGBP) maps. These maps are dissected and histograms of patterns formed for each segment. The histograms for all LGBP maps are sequentially concatenated for the final LGBPHS representation. 91

2.19 Face recognition using LBP (Ahonen et al. [2]). The face image (left) is pre-processed (middle) and segmented into multiple regions (right). LBP histograms are generated for each region and averaged over all samples for the individual. The resulting average histograms are concatenated to derive a descriptor for the individual. 92

2.20 The combination of Gabor wavelets with LBP for face recognition (Tan and Triggs [216]). The method involved the use of dimensionality reduction via PCA on feature vectors for each method individually, followed by concatenation for fusion and the application of Kernel Discriminative Common Vectors (KCDV) for the extraction of optimally discriminant nonlinear features. 93

2.21 Spatial-Featural Volume for the LBP features of an image. Each slice constitutes the binary-thresholded pixel from a specific position from the surround (e.g. 8 slices for an $N = 8$ LBP derived from 3x3 pixel regions surrounding each pixel). Previous methods encode all slices simultaneously without considering those which are most discriminative for classification. 94

3.1 An example of modelling the colour distribution of a multi-coloured object in Hue-Saturation space. Top: a multi-coloured object (a drinks can). Middle: its colour histogram (polar coordinates superimposed onto a Cartesian grid). It can be noted that since histograms are non-parametric, such a representation is only viable when a large amount of data is available. Bottom: its Gaussian mixture model. The mixture components are shown as elliptical contours of equal probability. 104

3.2 Application of the IMOS algorithm for generating a Gaussian mixture for the colours of the drinks can shown in Figure 3.1. Part (a) illustrates six iterations of the process, with each pair of images showing EM convergence followed by the splitting of a component. Part (b) shows the final seven component model (left) and the resulting probability density function with brighter regions corresponding to higher probabilities (middle). Finally, a histogram of the training data in polar coordinates is shown superimposed onto a Cartesian grid (right). 108

3.3 A face is tracked against a cluttered background by an active camera which pans, tilts and zooms. 114

3.4 Colour mixture models of a multi-coloured object (person model) and the context (scene model). The first row shows the data used to build the foreground (person) and the background (laboratory) models. The second row illustrates the probability density estimated from mixture models for the object foreground and scene background. The rightmost image is the combined posterior density in the HS colour space. Here the "bright" regions represent foreground whilst the "dark" regions give the background. The "grey" areas are regions of uncertainty. 115

3.5 Illustration of foreground-background classification-based pixel filtering for tracking. The top row outlines the tracked region for segmentation and the second row illustrates superimposition onto an alternative sequence. 116

3.6 Five frames from a sequence in which a face was tracked using a non-adaptive model. The apparent colour of the face changes due to (i) varying illumination and (ii) the camera's auto-iris mechanism which adjusts to bright exterior light. . 116

3.7 The sequence depicted in *Figure 3.6* tracked with an adaptive colour model. Here, the model adapts to cope with the change in apparent colour. 117

3.8 (a) Frames 35, 45, 55, 65 and 75 from a sequence. There is strong directional and exterior illumination. The walls have a fleshy tone. At around frame 55, the subject rapidly approaches the camera which is situated in a doorway, resulting in rapid changes in illumination, scale and auto-iris parameters. This can be seen in plot (b) which shows the 3D plot of the hue-saturation distribution over time. In (a), the model was allowed to adapt in every frame, resulting in failure at around frame 60. Plot (b) illustrates the use of selective adaptation. Plot (c) shows the normalised log-likelihood measurements and the adaptation threshold. 118

4.1 Scenario A Tracker Output: The centre-surround box shows tracker localisation for the object, with the center box denoting estimated object position and the surround region the area from which background samples are gathered. The dashed blue box indicates manually labelled ground truth. The bottom-left of each output frame depicts the tracked region zoomed in for clarity. See Figure 4.2 for corresponding confidence maps. 138

4.2 Scenario A Confidence Maps: The confidence maps correspond with the bounding box regions shown in the frames of Figure 4.1. The numbers below each image are the rounded sums of the confidences of pixels in the surround region. AFR generally shows cleaner, tighter confidence maps than both Collins et al. and Ensemble Tracking (ET) with the most salient part of the object emphasised more. The SemiBoost tracker (ST) shows sparser maps after frame 20. However, adding MSFM to AFR improves the AFR maps and further increases emphasis on the salient object region consistently for all frames, demonstrating the greater specificity of the adaptive models for AFR+MSFM. 139

4.3 Scenario A feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. Only features with non-zero weights are included. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner confidence maps (see Figure 4.2). The SemiBoost tracker (ST) showed the least variation in feature selection. From frame 20 onwards, the same single feature was effectively selected for classification, with all other features weighted zero. 140

4.4 Scenario A localisation errors in pixel distance between each of the trackers and manually-labelled ground truth. All trackers performed well for this scenario, with only Ensemble Tracking deviating in the last couple of frames. 141

4.5 Scenario B, Example 1 Tracker Output: Severe brightness change. Frame 5875 is gamma corrected for reader clarity. Brightness was reduced by subtracting from original pixel values until around halfway through the sequence when it was increased again (from left to right respectively, images show 100%, 31%, 2%, 48% and 112% of original average pixel values). The centre-surround box shows tracker localisation for the object. The dashed blue box indicates manually labelled ground truth. Collins et al., Ensemble Tracking (ET) and the SemiBoost tracker (ST) all failed to maintain tracking. AFR managed to track for longer but failed when the scene became extremely dark. AFR+MFSM was able to adapt to track the person's bag which was still sufficiently salient, until the person became strongly visible enough to be recaptured by the tracker for the remainder of the sequence. See Figure 4.6 for corresponding confidence maps. 143

4.6 Scenario B, Example 1 Confidence Maps: Severe brightness change. The numbers below each image are the rounded sums of the confidences of pixels in the surround region. Features selected by AFR (when locked on) resulted in improved confidence maps and pixel classification over Collins et al. and Ensemble Tracking, with emphasis on the most salient object region. AFR+MSFM improved on this further still. The SemiBoost (ST) tracker showed clean maps initially but failed earliest of all. 144

4.7 Scenario B, Example 1 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner confidence maps (see Figure 4.6). The SemiBoost tracker (ST) showed least variation in feature ranking, but the tracker failed the earliest. 145

4.8 Scenario B, Example 1 localisation errors in pixel distance between each tracker and manually-labelled ground truth. The SemiBoost tracker (ST) failed near the beginning of the sequence whereas Collins et al. and Ensemble Tracking (ET) were unable to cope with the darkest portion of the sequence and lost track. AFR failed shortly afterwards. However, AFR+MFSM was able to latch onto a still-salient part of the target object (a carrier bag) until sufficient brightness was restored for the tracker to reacquire the target object around frame 5950. 146

4.9 Recovery of AFR+MSFM after distraction. (a) Selected frames with ground-truth (dashed blue box) and tracker output (yellow box) superimposed. (b) Error in tracking position relative to ground-truth. (c) Most dominant feature from frame 5850 onwards. Shortly after frame 5860 the person lost saliency due to severe low lighting and the dominant feature changed from 16 to switching between 35 and 3, as can be seen from Plot (c). However, from around frame 5880, the tracker was able to attach itself to a still-salient bag being carried by the person, which was well characterised by feature 12 at that brightness range. As scene brightness returned, feature 3 reasserted itself around frame 5930. Around Frame 5950 the person became visible enough for the tracker to quickly reattach itself via the previously dominant feature 16, whose reference model asserted itself to reacquire the target object. From this point the bag was also consistently emphasised via feature 12, since via MSFM its representation was strengthened during the darker period, with the result that both relevant features (16 and 12) reinforced each other in characterising the tracked person. 148

4.10 Scenario B, Example 2 Tracker Output: Severe illumination change. The red and blue pixel values were modified (from left to right respectively) to 100%, 64%, 39%, 74% and 107% (for red) and 100%, 118%, 128%, 115% and 98% (for blue) of the original average pixel values. The centre-surround box shows tracker localisation for the object. The dashed blue box indicates manually labelled ground truth. Ensemble Tracking (ET) and the SemiBoost tracker (ST) both failed around the frame 5850 mark, with Collins et al. becoming distracted around 100 frames later. AFR and AFR+MFSM both maintained track for the duration of the sequence. See Figure 4.11 for corresponding confidence maps. . . 150

4.11 Scenario B, Example 2 Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. While the SemiBoost tracker showed a clean confidence map in the first frame before failing, the AFR and AFR+MFSM trackers showed the most useful confidence maps consistently throughout the sequence with the target object more strongly emphasised. The adaptive reference models of AFR+MFSM helped to further strengthen the most salient parts of the target object. 151

4.12 Scenario B, Example 2 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner, more relevant confidence maps (see Figure 4.11). The SemiBoost tracker (ST) showed a similar rigidity as for previous experiments, but this did not translate to tracking accuracy in practice. 152

4.13 Scenario B, Example 2 localisation errors in pixel distance between each tracker and manually-labelled ground truth. Ensemble Tracking (ET) and the SemiBoost tracker (ST) both failed before the halfway point of the sequence, whereas Collins et al. failed lost track around frame 5925. Both AFR and AFR+MFSM were able to maintain tracking for the duration of the sequence. 153

4.14 Scenario C Tracker Output: Tracking under occlusion. The dashed dark blue box shows ground truth. Collins et al., Ensemble Tracking (ET) and the SemiBoost tracker (ST) failed within a few frames, whereas AFR and AFR+MSFM continued to track successfully. AFR began to fail around Frame 52 due to occlusion (see Figure 4.17) but AFR+MSFM resisted distraction due to up-to-date feature reference models. 154

4.15 Scenario C Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. The top three rows represent trackers that failed early and so expectedly depict noisy or empty maps bearing no relevance to the tracking target. AFR maps were clean and representative up until failure, whereas AFR+MFSM again improved on AFR by further reducing noise and consequently improving actual tracking performance. 155

4.16 Scenario C feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than Ensemble Tracking (ET), with the AFR+MFSM again further improving on AFR alone. This resulted in cleaner confidence maps (see Figure 4.15). Collins et al. showed less fluctuation than normal, but this is attributable to its attachment to a relatively static portion of the background following tracking failure. The SemiBoost tracker (ST) selected the same single feature for the whole sequence, but again this was largely due to tracking failure occurring at the beginning of the sequence. 157

4.17 Scenario C localisation errors in pixel distance between each tracker and manually-labelled ground truth. Collins et al., Ensemble Tracking (ET) and the SemiBoost tracker (ST) failed almost instantly, whereas AFR was able to track the target before being distracted by a significant moving occluder shortly after frame 50. AFR+MFSM maintained a lock on the target for the duration of the sequence. . . 158

4.18 Scenario D, Example 1 Tracker Output: Tracking under occlusion and lighting change. Frames 80 and 120 were gamma corrected for reader clarity. Brightness was reduced by subtracting from original pixel values until around halfway through the sequence when it was increased again (from left to right respectively, images show 100%, 48%, 18%, 20% and 51% of original average pixel values). The dashed dark blue box shows ground truth. Collins et al. and Ensemble Tracking (ET) failed around frame 90, whereas the SemiBoost tracker (ST), AFR and AFR+MSFM all continued to maintain a lock successfully for the duration of the sequence. 160

4.19 Scenario D, Example 1 Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. The SemiBoost tracker (ST), AFR and AFR+MFSM all showed the most consistently representative confidence maps, with MFSM again showing its ability so support AFR in improving pixel classification. The SemiBoost tracker showed the cleanest maps as a result of its lack of flexibility in feature ranking (see Figure 4.20). 161

4.20 Scenario D, Example 1 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner confidence maps (see Figure 4.19), with AFR+MFSM again demonstrating meaningful improvement over AFR alone. The SemiBoost tracker (ST) again showed significant rigidity in feature ranking, which in this example proved adequate for tracking to succeed. 162

4.21 Scenario D, Example 1 localisation errors in pixel distance between each tracker and manually-labelled ground truth. Collins et al. and Ensemble Tracking (ET) were both distracted around frame 90, whereas the SemiBoost tracker (ST), AFR and AFR+MFSM all succeeded in tracking throughout the sequence. 163

4.22 Scenario D, Example 2 Tracker Output: Tracking under occlusion and lighting change. The dashed dark blue box shows ground truth. The red and blue pixel values were modified (from left to right respectively) to 100%, 70%, 46%, 40% and 39% (for red) and 100%, 114%, 128%, 139% and 135% (for blue) of the original average pixel values. The SemiBoost tracker (ST) failed within ten frames while the others continued to track. Collins et al., Ensemble Tracking (ET) and AFR all became distracted around frame 95, with AFR recovering within fifteen frames and Collins et al. recovering within thirty frames. AFR+MSFM stayed locked onto the target for the duration of the sequence. See Figure 4.25 for the ground truth plot. 164

4.23 Scenario D, Example 2 Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. Due to early failure, the SemiBoost tracker (ST) showed the least relevant confidence maps overall, with Ensemble Tracking (ET) also noisy following an unrecoverable failure around frame 95. As has been consistently demonstrated, MSFM helped to support AFR by reducing noise and improving tracking performance. Collins et al. showed a very similar quality in its confidence maps for this sequence although tracking performance was not up to the consistency of AFR and AFR+MSFM. 165

4.24 Scenario D, Example 2 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. As for all previous experiments, AFR+MSFM added further control to the feature ranking and selection statistics of AFR alone which in turn translated into actual performance (see Figure 4.25). The SemiBoost tracker (ST) again showed significant rigidity in feature ranking but failed early on. The statistics for Collins et al. and Ensemble Tracking (ET) showed a less haphazard characteristic than normal, but neither were able to track as consistently as AFR or AFR+MSFM. 167

4.25 Scenario D, Example 2 localisation errors in pixel distance between each tracker and manually-labelled ground truth. The SemiBoost tracker (ST) fails very early, whereas Ensemble Tracking (ET) continues up to around frame 95 where it fails irreversibly. Both Collins et al. and AFR become distracted by the same proximal distractor around the same time, with the latter recovering sooner than the former. Only AFR+MSFM was able to maintain a lock for the duration of the sequence. . 168

5.1 Strategies for LBP surround sampling. (a) The intensity samples used for thresholding are simply the pixels $x_0, ..., x_7$ from the 3x3 grid surrounding the centre pixel x_c. (b) Samples are taken by sub-pixel sampling on a circle at a fixed distance before thresholding. 173

5.2 An example of a fixed pre-determined sampling strategy for a multipredicate LBP. Each radius represents a spatial arrangement from which image samples are taken and corresponding textons derived for encoding (see Equation 5.1). The dashed lines indicate the samples which are considered jointly. As such, each radius is considered separately from the others, with the result that all scales are statistically disjoint from each other. 174

5.3 An example of a learned sampling strategy for a Multiscale Selected Local Binary Features (MSLBF) operator. A feature selection step chooses the most useful textons which are then used to form sparse, highly discriminative models with distinct spatial topologies which are fully coupled across scales. This figure shows three scales although any number of scales and corresponding samples may be added to a pool of textons for selection. 178

5.4 Examples from the Outex_00000 texture suite. 186

5.5 Average histogram separation per class for LBP and MSLBF models generated by BHIM, CMIM and AdaBoost. The mean histogram distance for each class for Outex_00000 is plotted. MSLBF+BHIM has significantly larger between-class distances. 188

5.6 Examples from the ORL face database demonstrating within-class variations of appearance, lighting and/or pose. 189

5.7 Average histogram separation for LBP and MSLBF models for the ORL face database. BHIM, CMIM and AdaBoost were compared for selecting features for MSLBF models. As with textures, the BHIM features show larger histogram distances. 191

5.8 Comparison of BHIM, CMIM and AdaBoost plotted over varying feature pool sizes and averaged over 100 random two-class datasets containing 12 embedded features with each class represented by 10000 samples. The plots denote; (a) average remaining entropy of the class variable given the selected features, (b) average proportion of selected features that match the embedded features and (c) the time required to select 12 features. BHIM was clearly more successful at finding the embedded features as shown in plots (a) and (b), without being influenced by large numbers of distractors in the pool. Computation time was linear with the size of the feature pool. 198

5.9 Comparison of BHIM, CMIM and AdaBoost plotted over varying numbers of training samples per class and averaged over 100 random two-class datasets with 12 embedded features. The plots denote; (a) average remaining entropy of the class variable given the selected features, (b) average proportion of selected features that match the embedded features and (c) the time required to select feature subsets. As for Figure 5.8, BHIM was more successful at finding the embedded features without being affected by the number of training samples. Computation time was linear with the size of the training set. 199

5.10 Comparison of BHIM, CMIM and AdaBoost plotted over varying numbers of features to select and averaged over 100 random two-class datasets containing 12 embedded features with each class represented by 10000 samples. The plots denote; (a) average remaining entropy of the class variable given the selected features, (b) average proportion of selected features that match the embedded features and (c) the time required to select feature subsets. Here, it can be seen in (a) that as more and more features were recovered by BHIM, the closer the remaining entropy dropped to that of the embedded set. This is to be expected since the reference entropy was computed from all 12 embedded features. Plot (b) illustrates that the features selected by BHIM were always strongly correlated with those embedded. Computation time for BHIM was exponential with the number of features to select as opposed to linear for CMIM and AdaBoost. However, total time taken to recover all 12 features was not excessively higher. . 200

List of Tables

5.1 Classification of Outex 00000 database which contains variations of canvas, tile and carpet type textures. The multipredicate $LBP^{riu2}_{8,1} + ^{riu2}_{16,3} + ^{riu2}_{24,5}$ classifier is generated from training data comprising predicates at 1, 3 and 5 pixels radius with 8, 16 and 24 samples per predicate. The MSLBF classifiers were generated with three different feature selection algorithms on a larger training set comprising six predicates at 1, 2.5, 4, 5.5, 7 and 8.5 pixels radius with 8, 16, 24, 32, 40 and 48 samples respectively. The MSLBF results are given along with a value in brackets indicating the lowest number of features required to achieve the corresponding success rate for that class (up to a maximum of 8). 187

5.2 Overall success rate of the four classifiers with Outex_00000 along with the average number of features needed to gain the best scores shown in Table 5.1 per class. The LBP classifier is constructed from smaller sample areas than MSLBF. . 188

5.3 Overall success rate of the four classifiers with the ORL database along with the average number of bits needed to gain the best scores. The MSLBF classifiers were constructed from the same sample regions as the LBP classifier. 189

5.4 Classification results for the ORL face database. An $LBP^{u2}_{8,1} + ^{u2}_{16,3} + ^{u2}_{24,5}$ classifier is generated from training data comprising predicates at 1, 3 and 5 pixels radius with 8, 16 and 24 samples per predicate. The MSLBF classifiers were generated with three different feature selection algorithms on the same data set. The numbers in brackets indicate the lowest number of features required to achieve the corresponding success rate for that class (up to a maximum of 8). 190

List of Algorithms

2.1 The AdaBoost algorithm for binary classification tasks (Freund and Schapire [62]). The procedure involves training an ensemble of classifiers in sequence, each one on training data re-weighted according to the performance of those classifiers already trained. Consequently, "hard" examples are given greater focus of attention with the result that trained "weak" classifiers taken together perform more strongly. This method may also be used as a wrapper for feature selection (refer to text). 62

2.2 The Ensemble Tracking algorithm (Avidan [4]). An ensemble of weak classifiers is trained by AdaBoost. At each new frame, new weak classifiers are trained and used to replace the weakest ones in the current ensemble. 80

3.1 The Iterative Model Order Selection (IMOS) algorithm for Gaussian mixture models. The model is initialised with a single component. Thereafter, the algorithm repeatedly splits the component with the lowest responsibility for the validation set (Equations 3.4 and 3.5) to create a new higher-order model, applies Expectation-Maximisation to fit the new model to the training set and monitors the log-likelihood of the validation set with respect to the new model. At this stage, a lower log-likelihood than the previous one is considered as an indication of overfitting and the previous model (corresponding to a peak in the likelihood progression) returned as the final result. 107

4.1 The Adaptive Multi-feature Association (AMA) algorithm. Tracking initialisation is assumed to be provided along with the first frame. All priors are initialised to 0.5 and these values used for the second frame; they are updated for subsequent frames using Equations 4.7, 4.8, 4.15 and 4.16. Reference models are then updated using Equations 4.9 and 4.12 (all features if the first frame, only the top N if subsequent). All features are then ranked using Equations 4.1, 4.4 and 4.5. The next frame is then collected and confidence maps computed using the log-likelihood ratio (Equation 4.17). Finally, mean-shift is applied to estimate new object position and the cycle restarts. 134

5.1 The Binary Histogram Intersection Minimisation (BHIM) algorithm for selecting K features from two binary data sets $\mathcal{P}$ and $\mathcal{Q}$ containing J features. The function $decval(\zeta, B)$ returns a set of decimal values for each training sample in class ζ generated from the joint values across the features in B. The function $S(\mathbf{w}, \mathcal{P}, \mathcal{Q}, B, j)$ computes the weighted binary histogram distance corresponding to Equation 5.8. 182

5.2 The Multiscale Selected Local Binary Features (MSLBF) classification algorithm. $genhist(I, B_{\mathcal{P},\mathcal{Q}})$ is a function returning the histogram for input data I given the features $B_{\mathcal{P},\mathcal{Q}}$ specific to the pair of classes $\mathcal{P}$ and $\mathcal{Q}$. The $HI\{h(X|\Phi), h(X|\Xi)\}$ function computes the histogram intersection between two histograms $h(X|\Phi)$ and $h(X|\Xi)$. 184

List of Notations

Symbol	Definition
2^S	Set of all subsets of S
A	Sample set
b	Feature index
B	Background; Set of selected features; Sample set
c	Confidence map; binary classifier; target function
C	Weighted averaged confidence map; set of concepts
d	Discriminability
D	Recursive sum; distribution
$D_{KL}(\cdot\|\|\cdot)$	Kullback-Liebler divergence between two distributions
F	Foreground
$\mathcal{F}$	Feature space
h	Bounding box height
$h(\cdot)$	Sample histogram for random variable X
$H(\cdot)$	Entropy of random variable X
$I(\cdot;\cdot)$	Mutual information between two random variables
I_t	Frame t
i	Index variable
j	Index variable
J	Size variable
K	Size variable
k	Index variable
l	Index variable
L	Temporal window; classifier
$\mathcal{L}(\cdot;\cdot)$	Log-likelihood of data given a model

Symbol	Definition
m	Dimensionality
$\mathbf{m}$	2D spatial mean
M	Feature Reference Model
$\mathbf{M}$	Subset
$\hat{\mathbf{M}}$	Optimal subset
n	Dimensionality
N	Size variable
O	Object
$p(\cdot)$	Probability density
$\mathbf{p}$	Set of parameters
P	Number of LBP textons
$P(\cdot)$	Probability
$\mathcal{P}$	Dataset
$\mathcal{Q}$	Dataset
r	Feature ranking score
R	Spatial region; set of real numbers; LBP radius
$R(\cdot)$	Pearson Correlation Coefficient
$\mathbf{s}$	Score value
S	Sample set
$S(\cdot)$	Weighted binary histogram distance
T	Threshold
$T(\cdot)$	Template
t	Frame index; texton
U	Random variable
u	Independence
$\mathbf{u}$	Eigenvector
V	Validation set; random variable
w	Weighting value; decimal value; bounding box width

Symbol	Definition
$\mathbf{w}$	Basis function
W	Random variable
$\mathbf{W}(\cdot;\cdot)$	Warping of points given some parameters
x	Value of a random variable X
$\mathbf{x}$	Value of a random vector $\mathbf{X}$
X	Random variable
$\mathbf{X}$	Random vector
y	Value of a random variable Y
Y	Random variable
$Y \perp\!\!\!\perp X$	Y is statistically independent of X
β	Scaling factor
ε	Threshold
$\varepsilon(\cdot)$	Probability of misclassification
ζ	Class label
$\zeta(\cdot)$	Class of data point
θ	Gaussian mixture component parameter
κ	Normalisation constant
λ	Eigenvalue
$\boldsymbol{\mu}$	Mean
ξ	Pixel coordinates
Ξ	Model
ρ	Correlation
σ	2D spatial standard deviation
$\boldsymbol{\Sigma}$	Covariance matrix
υ	Bounding box
Φ	Model
ψ	Sum of posterior probabilities
$\Psi(\cdot)$	Function

List of Abbreviations

Abbreviation	Definition
AdaBoost	Adaptive Boosting
AFR	Attribute-based Feature Ranking
AMA	Adaptive Multi-feature Association
BHIM	Binary Histogram Intersection Minimisation
BLBP	Bayesian Local Binary Patterns
CCA	Canonical Correlation Analysis
CMIB	Conditional Mutual Information based Boosting
CMIM	Conditional Mutual Information Maximisation
EM	Expectation-Maximisation
ET	Ensemble Tracking
FLS	Filtering, Labeling and Statistic
GMP	Gabor Magnitude Picture
HMM	Hidden Markov Model
HoG	Histogram of Gradients
HSV	Hue-Saturation-Value
ILBP	Improved Local Binary Patterns
IMOS	Iterative Model Order Selection
JPDAF	Joint Probabilistic Data Association Filter
KDCV	Kernel Discriminative Common Vectors
LBP	Local Binary Patterns
LBPH	Local Binary Pattern Histogram
LDA	Linear Discriminant Analysis

Abbreviation	Definition
LGBP	Local Gabor Binary Pattern
LGBPHS	Local Gabor Binary Pattern Histogram Sequence
LTP	Local Ternary Patterns
MAP	Maximum A-Posteriori
MB-LBP	Multi-scale Block Local Binary Patterns
MCMC	Markov-Chain Monte Carlo
MSFM	Multiple Selectively-adapted Feature Models
MSLBF	Multiscale Selected Local Binary Features
NLDA	Null-space Linear Discriminant Analysis
OCLBP	Opponent-Colour Local Binary Patterns
PCA	Principal Components Analysis
RGB	Red-Green-Blue
SFV	Spatial-Featural Volume
SIFT	Scale Invariant Feature Transform
ST	SemiBoost Tracking
V-LGBP	Volume-based Local Gabor Binary Patterns

Chapter 1

Introduction

Vision is at once both enigmatic and yet perhaps one of the most taken for granted of the senses. It facilitates a wide array of tasks and yet for most the question of the complexities and sophistication that underlie this capability never arises. From the electric impulses of an array of photoreceptive cells we are ultimately able to survey our environment, navigate from one location to the next, find food, detect danger, recognise a family member, play football, appreciate art and many other activities without which our lives might be experientially less multifarious. Given this immense versatility of vision, it is unsurprising that the pursuit of artificial systems that embody its capabilities is widely undertaken. Such technology promises to enhance life in a variety of ways, from the automatic visual analysis of medical images for anomaly detection (e.g. [137, 9, 217], Figure 1.1) to replacing human drivers on the road (e.g. [220, 46, 50], Figure 1.2) to round-the-clock exhaustive surveillance of critical public areas for detecting suspicious or anomalous behaviour (e.g. [87, 252, 85], Figure 1.3). In the process of this drive, it is perhaps to be expected that sets of seemingly disparate visual phenomena will be found to be rooted in some common foundation, even if in a wide variety of manifestations.

1.1 Visual sampling in humans

The field of human psychophysics is concerned with experimentally exploring the underlying cognitive mechanisms responsible for certain human visual functions, such as the ability of people to perceive surfaces as uniformly bright under changing illuminant or the perception of two-dimensional images in a three-dimensional context. One area of interest involves the mecha-

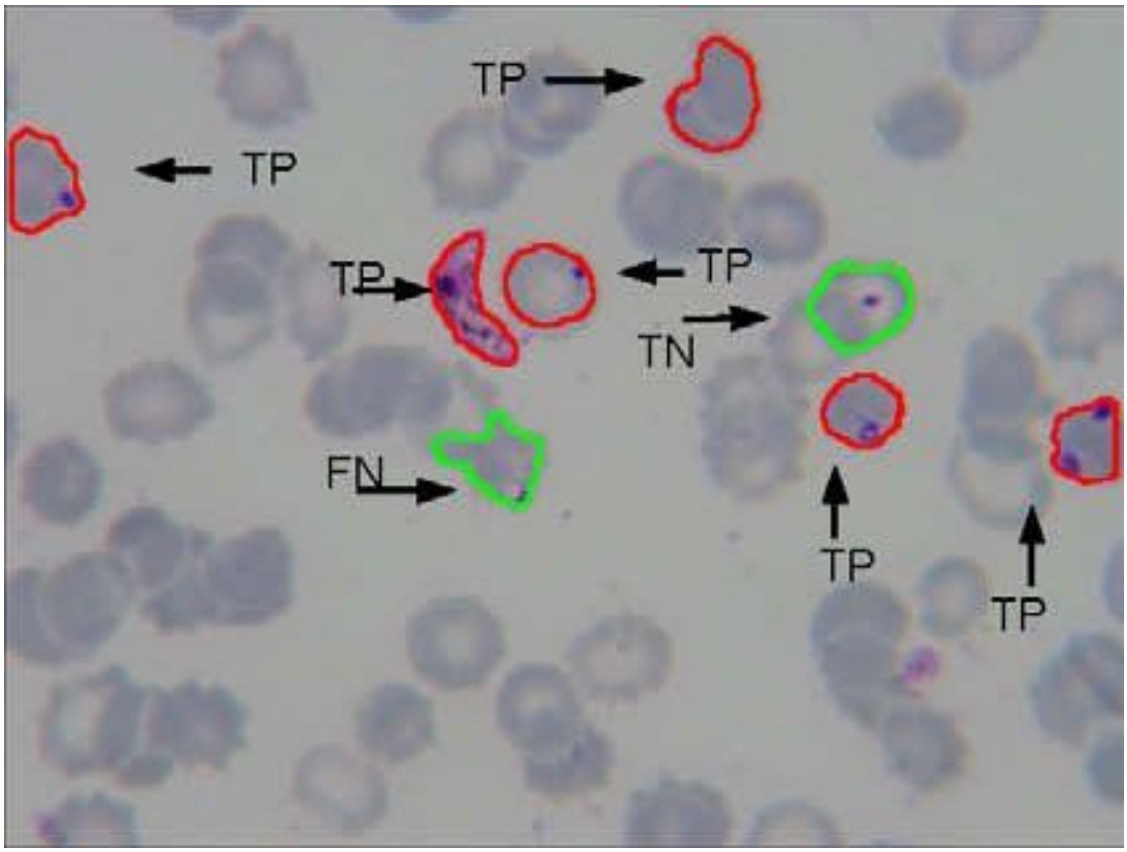


Figure 1.1: Example of automated segmentation and classification of malarial parasites in a blood specimen. True positives (TP), true negatives (TN) and false negatives (FN) are highlighted. (Tek et al. [217]).

nisms of *visual attention* and the factors influencing the direction of attention of humans such as limited (processing) capacity and the consequential efficiency and efficacy with which various visual tasks may be performed (e.g. [159, 45, 38]). Models of visual attention tend to fall into two complementary categories [54]; visual *search* (e.g. [49, 109]) and visual *sampling* (e.g. [160, 157]). By standard definitions, the former essentially involves locating a specified object or set of objects whose presence or position is unknown in the visual environment (e.g. finding a set of keys in a cluttered room) and the latter involves determining the state of objects or events at predetermined places to facilitate a specific visual task (e.g. scanning gauges on the dashboard when driving on the motorway). However, in a more general context, at a low-level both of these may be viewed as intrinsically relying on an ability to perform associations between stimuli and appropriate internal representations or models.

There are several factors that influence the human sampling of the visual field during a visual task. For example, Kundel [104] compared the visual attention patterns for laypersons and

Figure 1.2: Detection of road boundaries and road junctions for automated vehicle navigation. Detected roadside boundaries (the thick black and white lines) are used to estimate the trajectory of the road and the position/orientation of junctions (the thin black lines). (Ekinci et al. [50]).

radiologists in analysing chest x-rays, with the unsurprising result that the experts' distribution of foveal fixation areas coincided with those most likely to contain an anomaly. For drivers, Senders et al. [192] studied required visual sampling frequencies when visibility of the road was periodically disrupted and Wierwille [232] observed sampling patterns when drivers were simultaneously preoccupied with an in-vehicle task. Additionally, Hoffman et al. [84] studied the strategies employed by drivers for visually sampling the road during driving whilst simultaneously examining text messages on devices inside the vehicle. Such experiments consistently suggested that visual sampling of the road when distracted during driving is geared towards maintaining levels of uncertainty regarding the position of the vehicle relative to the road below a threshold. The efficiency and efficacy of the sampling process was in turn affected by factors such as the demands imposed by the distracting task and the level of experience of the driver. A more abstract study was done by Kvälseth [105] to explore visual sampling within an information-theoretic context for a Gaussian autoregressive process. His experiments suggested

Figure 1.3: Detection of anomalous road crossing ("jaywalking") in automated public CCTV analysis (Hospedales et al. [85]).

that sampling was geared towards maximising information gain when uncertainty rose above a certain point. Wickens et al. [231] investigated the visual sampling patterns of pilots in a cockpit environment whilst engaged in traffic detection. They derived a successful model that reflected the probability of areas of interest being scanned during a visual sampling process as a combination of the value of scanning such areas, the relevance of the information that could be gained there and the bandwidth (or amount of information) that could be collected during the scan. They also explored the importance of salience of areas of interest and the amount of effort required to scan them. Studies such as these and others serve to highlight the role of various factors in visual sampling such as limited cognitive capacity, the task being performed, experience, information gain, efficiency and salience, amongst others. Consequently, visual sampling may be viewed as inherently dynamic, adaptive and context-dependent.

1.2 Visual sampling in computer vision

Human psychophysical explorations of visual sampling, such as the work described above, are on the whole rather high-level and abstract. The question arises as to the context in which computer vision algorithms could or should perform visual sampling. While concepts derived from visual attention research such as limited capacity, efficiency and information gain may be directly related in a concrete way to computer vision, ideas borrowed from statistics and which are widely employed by computer vision algorithms serve to place a slightly lower-level, less abstract definition on the notion of visual sampling. In statistics, the idea of sampling in general is geared towards the collection of samples which satisfy some criterion, such as being *representative* of a population. The representative sample may then be used to infer some properties regarding the overall population. The process of collecting such samples is a *sampling strategy*. Sampling strategies may take several forms with the end result that a set of appropriate samples has been collected. On one hand, samples may be gathered for the purpose of minimising the amount of data that must be stored without losing relevant information. Subsampling an image to reduce its dimensions whilst retaining the overall forms reflected therein is a simple example of this approach (e.g. see Figure 1.4). A more sophisticated example is the recent method of *compressive sampling* (Candes and Wakin [19]), which is capable of compressing images into extremely sparse basis representations and reconstructing them in high detail. On the other hand, samples may need to be scrutinised and accepted or rejected in order to meet some criterion, such as finding pixels in an image which belong to a specific object in the scene (e.g. see Figure 1.5). In this case, the sampling strategy essentially takes the form of a *filtering* process, since the samples are effectively being drawn from a combination of populations, only one of which is of interest. This resulting set of samples may then be taken as representative of the corresponding object (population) and may be used to estimate some desired properties of that object such as its position, distance or pose. At a more abstract level, strategies have been carefully constructed to facilitate sampling from probability densities that cannot be directly accessed, in particular Markov-Chain Monte Carlo (MCMC) techniques such as Gibbs sampling [68] or the filtering-based sampling strategy used by the more general Metropolis-Hastings algorithm [146] which involves accepting or rejecting samples according to a ratio value.

For computer vision tasks, the filtering-based approach to sampling is of particular interest. The foundation of many application areas such as tracking, segmentation and recognition con-

Figure 1.4: Spatially regular pixel sampling of an image to reduce the required storage while maintaining the fundamental physical forms within. The right image is the same as the left but subsampled to contain only 1% of the pixel data. While there is less detail, the overall spatial shapes and relationships across the entire image are maintained.

Figure 1.5: Selective pixel sampling based on object membership. The right image retains only those pixels which are classified as belonging to the red portion of the rose in the left image. In this case, the sampling strategy is to accept or reject pixels based on the probabilistic likelihood of having been generated by a model representing either the red part of the flower or the rest of the image.

sists of reliably establishing the level of association between image samples and internal models of some type. Ultimately, this results in the allocation of values to pixels or derived transformations of pixels which indicate various degrees of confidence in the association and from which assignment decisions can be made. Such considerations may be made for pixels either individually (e.g. background modelling [206]) or in conjunction with each other (e.g. template tracking [135, 149] or background modelling [185]). Consequently, it might be argued that, most, if not all, computer vision tasks are rooted in a filtering-based sampling strategy of some kind. The task then is to design effective sampling strategies that satisfy the requirements of being adaptive, dynamically contextually-relevant and efficient for a given visual application. However,

many computer vision methodologies are severely lacking in catering for this philosophy, either through the use of static a priori models or the use of inflexible algorithms based on limited assumptions that prevent their effectiveness in real-world situations. A key issue related to this is the question of *averaging* vs *selection* as filtering approaches. Most computer vision methodologies have traditionally used optimisation techniques to derive models from sample data such that the samples are effectively interpolated or *averaged* over. While outliers in such data tend to be downweighted, they can still impact the efficacy of a model to various degrees. This becomes even more acutely problematic in model update situations where errors due to outliers can build over time, leading to *model drift*. While this may be the best approach in many cases, in many others it may be more appropriate to perform *selection* while completely discarding those samples not selected.

1.3 Object association

Object association is concerned with the matching of pieces of image evidence with specific objects of focus. This capability underpins a range of visual tasks such as tracking and recognition. Tracking involves the use of *data association* techniques [5, 39] to combine *prediction* and *verification* using sensory measurements to update estimates of target state (such as spatial position) over time. If sensory measurements are obtained from more than one source (such as radar and infrared signals), *data fusion* methods (see Hall and Llinas [78]) may be employed to combine them when updating states. More specifically for visual tracking, given a predicted position, the verification process generally involves separating foreground (the object of interest) from background (the surrounding area) and making a final judgement about target position. This amounts to a strategy that seeks to extract object-representative samples through a two-step sample filtering process: (a) narrowing the search through prediction; and (b) selecting samples through verification using object models (e.g. [88]). Accurate segmentation requires taking this to a certain degree of pixel-level precision; in essence, recovering all samples from a specific population. In real-world scenarios, tracking must often take place under difficult conditions where lighting, pose, clutter and multiple distractors with a similar appearance to the target can obfuscate the process (see Figure 1.6). Due to the inherently dynamic, complex nature of this problem, models often exhibit extreme transience in relevance, strongly discriminant at one moment and becoming impotent the next. As such, flexibility is required in the verification filtering process in order to

properly maintain the models that guide it. More specifically, we can envisage two components to maintaining models:

1. Adapting models to cope with changes in appearance that may result from factors such as changes in lighting. For example, distributions of colour will generally shift in feature space when conditions of illumination change over time. Empirical techniques for such an approach will generally involve the periodic sampling of object pixels to provide cues for adaptation; consequently, a crucial consideration is the prevention of *model drift*, which can occur through flawed or inadequate sampling strategies that introduce "contaminants" in the form of non-object pixels into the model. Such errors build over time, resulting in an eventual failure of the model to represent the tracking target.

2. Selecting the most appropriate features to use. For example, photometric features such as colour may be the most distinguishing characteristic when tracking a person wearing a red coat through a crowd of people wearing black, but if he or she then moves through a crowd of people wearing red, geometric features such as body shape or height may become more appropriate for making an association. Although some features may exhibit invariance under a relatively wide range of conditions, none are likely to be rigid against all of the variations experienced in everyday situations.

Previous methods such as those described by Collins et al. [29], Avidan [4] and Grabner et al. [72] go some way towards addressing the issue of selecting models with an inherently adaptive component that enables them to handle *short-term* changes in object appearance. However, they do not adequately address the problem of controlling *model relevance*; that is, the integrity of such models in reflecting object appearance at any given time. In attempting to restrict erroneous short-term changes which would over time render the models useless, they are made too restrictive in adequately adapting to *long-term* changes.

In contrast to tracking, object recognition is often performed by simply determining which population a set of already collected samples is likely to have been derived from rather than filtering samples taken from a mixture of populations to isolate those from a specific population. However, the general notion of spatially-selective sampling to increase the specificity of the samples being matched so as to improve discrimination also applies. For example, a person may not need an entire picture of an elephant in order to identify it - knowing where to look for

(a) (b)

(c) (d)

Figure 1.6: An example of a scenario for tracking a person (highlighted with a yellow arrow) through a crowd of people in a low-resolution, cluttered scene. Initialised in (a), the person moves across similarly coloured distractors (b), undergoes severe occlusion (c) and experiences chromatic change due to lighting fluctuations in (d). Such conditions put an insurmountable strain on features to stay representative of a target object during tracking to the exclusion of proximal distractors, suggesting a more appropriate strategy of dynamically selecting the most discriminative feature type at any given time as well as updating their corresponding models.

the most distinguishing characteristics is often sufficient as well as perhaps more *robust* under occlusion. Furthermore, as with tracking, the type of feature most useful is again fundamentally applicable - a flamingo is more likely to be pink than an elephant.

1.4 Approach

In our research, we place the emphasis on a conceptual and practical unification of visual tracking and recognition through their interpretation as being rooted in various forms of adaptive sampling strategy [138, 140, 139, 176, 177, 175, 178, 141, 174, 173]. In doing so, we extend previous work in several key areas to cater for the important requirements of efficiency, contextual relevance and adaptivity in these strategies (see Figure 1.7. A basis for the concepts presented herein is a formal extension of the 2D spatial domain of an image to the *featural* domain, which we call the Spatial-Featural Volume or SFV (Figure 1.8), analogous to the extension of three-dimensional space to the fourth dimension of time. Whereas a typical static image consists of a single type of feature sampled at regular 2D spatial intervals (e.g. greyscale values at each pixel location), an image more generally may be viewed as a volume with a third dimension indexing feature type and each "slice" along this dimension consisting of coordinate-indexed values for a specific feature. For example, three slices of the volume for an image may consist of red, green and blue coefficients of the RGB feature domain along with other slices corresponding to edge pixels, different colour spaces, local intensity histograms and any other type of pixel-referenced feature. Such features may either be retrieved directly (for example, by alternative sensory devices that sample the environment such as infra-red cameras or radar imagers) or derived through transformations of raw RGB pixel values. We interpret the tasks of object tracking and recognition as implementations of sampling strategies that dynamically filter pixel samples on-the-fly both in the spatial *and* the featural domain, given some set of features representing the latter; that is, in the same way that samples are filtered *spatially* according to their association with models, they are also filtered *featurally* according to their discriminative capacity. We focus our study on adaptive visual sampling in three areas of computer vision; object tracking, object association by classification and object recognition.

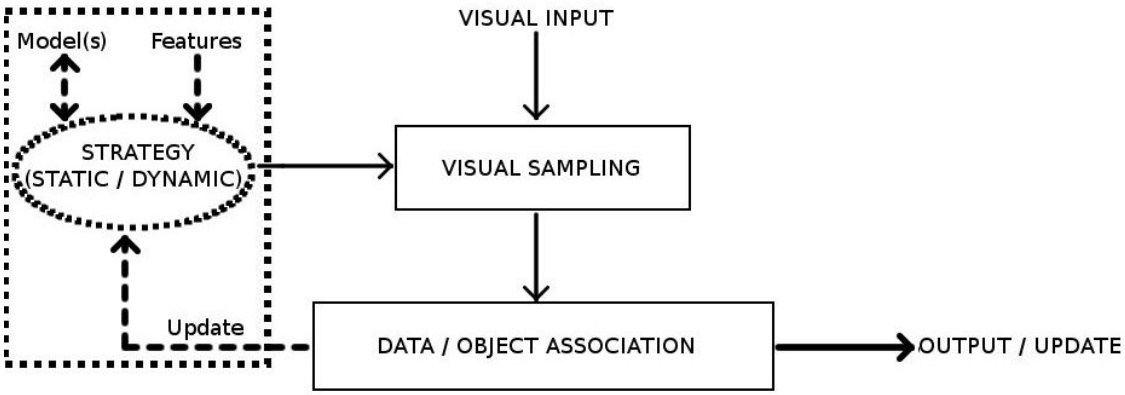


Figure 1.7: Visual tasks such as tracking and recognition (in both humans and machines) involve the visual sampling of input stimuli given a specific sampling strategy. Many computer vision sampling strategies are static both in terms of features and models used. The focus of this work is on developing flexible computer vision sampling strategies (dashed box) for tracking and object recognition which consist of feature selection and/or model adaptation to cater for the needs of contextual relevance and changing object appearance over time.

1.4.1 Colour feature sampling for tracking

A prime concern for any object association task being conducted over time is the continuing relevance of the object-representational model. This model forms the basis for an accurate filtering of pixels or features in order to derive a set representative of the object's population. In most real-world situations, dynamic conditions such as viewpoint changes and lighting fluctuations cause transience in the relevance of these models. This problem is particularly acute for tracking tasks in uncontrolled environments, where static object-representational models may lose their relevance frequently and rapidly. We would also argue that invariant features by and large do not exist, or at least are insufficiently representative of target objects without unique geometries. Consequently, such models need to be adapted accordingly; in other words, the sampling strategy should be flexible and dynamic.

We address this by studying adaptive statistical modelling of object features; specifically, colour values in Hue-Saturation space. These models are used for real-time and robust tracking of multi-colour objects under changing lighting conditions. Colour features have been used for a variety of tasks such as segmentation [201], tracking [138] and object recognition [134, 212]. We examine previous colour-based methods appropriate to this area and address specific limitations which are; (a) The use of non-parametric models (e.g. [212, 100]) which can be sensitive to small

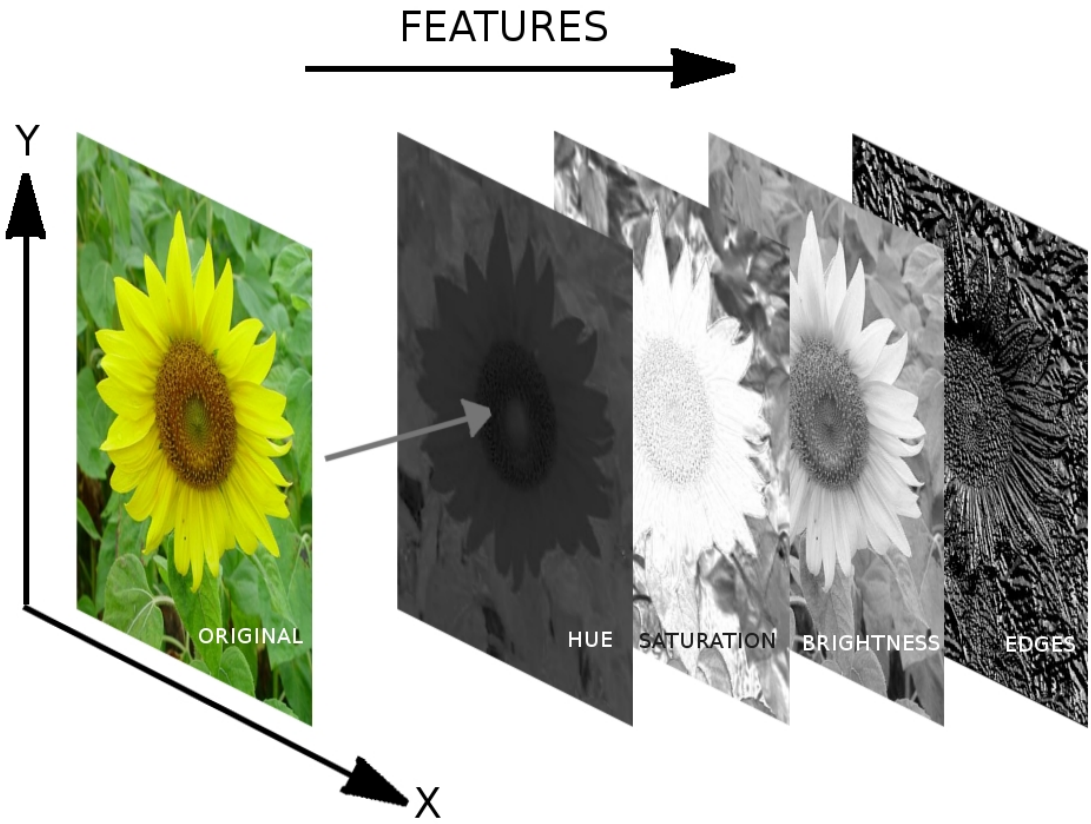


Figure 1.8: An example Spatial-Featural Volume for an image. Each "slice" constitutes a property or transformation of the original image, such as hue or edge-orientation at each pixel. Slices may themselves comprise multiple layers, such as the Histogram-of-Gradients HoG [41] descriptor comprising the statistics of oriented gradients in the fixed neighbourhood of each pixel. Vision tasks generally involve the evaluation of samples with the $2D$ x-y spatial domain but feature selection may also be viewed in this context as sampling from the third, "featural" dimension.

training sets and (b) The lack of a dynamic sampling strategy which prevents object models from maintaining relevance under changing conditions over time. In doing so we develop a sampling strategy framework for tracking based on pixel feature statistics and apply it to colour sampling in particular. This involves; (a) The use of semi-parametric Gaussian mixture models to capture multi-colour distributions. This has the benefit of enabling a more flexible selection of model order than histogram bin sizes as well as being better able to deal with small quantities of data; (b) An algorithm for automatically selecting the number of components of a Gaussian mixture model. This employs a cross-validation approach to incrementally add components and terminate when the model begins to overfit; (c) A fast filtering-based sampling strategy that employs

the model to sample from the image of a sequence and estimate object position and size. This operates in real-time on extremely modest hardware; (d) A Bayesian formulation for context-dependent pixel classification that more rigorously employs adaptive models of both object and background; (e) A mechanism to facilitate a dynamic sampling strategy by automatically adapting a mixture model to deal with colour changes caused by changing lighting conditions. Our approach also includes a more intelligent mechanism for detecting tracking failure and suspending adaptation in order to prevent model drift. We demonstrate the effectiveness of adaptive over static models on sequences depicting the tracking of faces against changing backgrounds under dynamic lighting conditions and undergoing partial occlusion with selective adaptation. We also demonstrate the Bayesian filtering formulation by tracking a multi-coloured torso with segmentation to illustrate the accuracy of pixel classification.

1.4.2 Multi-feature sampling for tracking

In addition to the effect of changes in appearance of an object on models used for sample filtering, the robustness of different *types* of features is highly context-dependent, with photometric and geometric environmental conditions inducing different complications. For example, colour is sensitive to changes in lighting whilst shape and texture may be drastically altered during pose transitions. Additionally, in cluttered scenes dynamic distractors can significantly affect the relevance of specific features over time, e.g. colour may perform adequately when a red target object is tracked against a non-red background but shape may be more discriminative when the target moves into a red-coloured area. Consequently, in the absence of truly robust features, a successful tracker will not only take measures to alleviate the problems of model-drift but also to utilise the features most likely to be useful at any given time.

We extend the sampling strategy for tracking described in Section 1.4.1 by incorporating an additional filtering step to address the issue of selecting appropriate features during tracking. Whilst previously filtering was performed purely in the spatial domain, here the notion is extended to another dimension, the *featural* domain of the Spatial-Featural Volume (SFV) as illustrated in Figure 1.8. This extra dimension constitutes an extra source of pixel samples, with each spatial image coordinate now indexing a pool of values derived from various transformations of the raw image colours. Previous work has addressed the issue of selection from this pool for tracking (e.g. [29, 4, 115, 72]) with the following limitations: (a) in choosing the most relevant features in each frame, features are ranked using metrics or boosting methods which

do not address issues of redundancy amongst those selected (see Section 2.1.2); and (b) model drift is inadequately addressed by using static reference models based on unrealistic assumptions of long-term relevance. Here we address these problems within a framework called *Adaptive Multi-Feature Association* (AMA) consisting of two components: (a) A more reliable feature ranking method called *Attribute-based Feature Ranking* (AFR) which consists of a combination of two computed attributes per feature to reduce redundancy; and (b) a mechanism called *Multiple Selectively-adaptive Feature Models* (MSFM) for maintaining multiple longer-term reference models that are selectively adapted on-line to avoid model drift. This forms an extension of the dynamic sampling strategy concept to multiple feature domains. Due to significant discontinuities in the visual appearance of objects, we consider the tracking problem as one of *object association* by classification. We demonstrate the effectiveness of this framework in challenging tracking scenarios depicting small target objects in cluttered environments undergoing severe lighting changes and occlusions and show that the use of adaptive reference models is a more effective approach to balancing short and long-term evidence to maintain relevance for the sampling strategy.

1.4.3 Local Binary Pattern feature sampling for recognition

Visual recognition, like visual tracking, finds its basis in the problem of object association. Although tracking is usually concerned with distinguishing between the foreground object being tracked and the background, this can naturally extend to discriminating between different foreground objects, as addressed by Song et al. [204] who employ object classification to perform disambiguation when tracking people in crowded scenes. In either case, there is a fundamental requirement for a sampling strategy that scrutinises the most appropriate parts of an object in order to both reliably and efficiently perform discrimination.

We consider the problem of object recognition using a sparse feature representation and in particular address some of the limitations of Local Binary Pattern (LBP) based recognition models. LBP methods have been used extensively for a huge range of applications, including texture discrimination [130, 155] (demonstrating excellent results and good robustness against rotation and global illumination changes), texture segmentation [172] and recognition of facial identity [2] and expression [53, 193, 195]. This paradigm involves representing classes of objects such as faces or textures by the joint statistical modelling of Boolean features yielded by thresholding samples from the surround of each pixel in corresponding images to capture local structure. We

examine the limitations of previous LBP methodologies and argue that they are borne from a fundamentally inflexible approach to modelling LBP statistics which: (a) either limit the spatial area over which models may capture information or otherwise average over potentially useful details; (b) incorporate possibly redundant information which wastes resources; (c) decouples statistics in an ad hoc manner; and (d) builds models in a spatially non-selective, context-agnostic way which further impacts on classification accuracy and betrays the inherent importance of adaptive visual sampling approaches to discrimination. We then propose a framework which solves all of these problems and can be added without modification to many existing LBP-type methods which involve modelling distributions of jointly encoded binary sequences such as [127, 118]. This framework involves: (a) a novel feature selection algorithm designed for binary data, called Binary Histogram Intersection Minimisation (BHIM), which is capable of finding stronger, less-redundant feature subsets than two state-of-the-art algorithms for binary feature selection; (b) the encoding of selected features to form distributions from context-dependent spatial topologies called Multiscale Selected Local Binary Features (MSLBF); and (c) the use of MSLBF models in a pairwise-coupling [82] scheme to enable the most appropriate samples to be used depending on the two classes being compared. We demonstrate the effectiveness of the framework over traditional LBP approaches in two specific recognition applications; texture classification and face recognition. Simultaneously, we show the improved descriptiveness of the feature selected by the BHIM algorithm over two established algorithms used for binary feature selection. We also present a third experiment that performs an extensive comparison of the three feature selection methods on synthetic data and shows the improved performance of BHIM without an excessive increase in computational cost.

1.5 Contributions

The main contributions of this thesis are:

1. We present a general spatial sampling strategy for tracking that explores adaptive statistical modelling of colour feature distributions to facilitate real-time and robust performance. We apply it to dynamically and selectively maintain the relevance of a colour model during a tracking task to improve reliability under changing lighting conditions and backgrounds. The framework consists of five components:

   (a) Gaussian mixture models for semi-parametric modelling of the colour distributions

of multi-colour objects;

(b) An Iterative Model Order Selection (IMOS) algorithm that uses cross-validation for automatically determining the number of components for a Gaussian mixture given a sample set of object colours, a procedure normally performed in an *ad hoc* manner;

(c) A sampling strategy for performing fast tracking using colour models;

(d) A Bayesian formulation enabling models of object and the environment to be employed together in filtering samples by discrimination;

(e) An adaptive mechanism to enable colour models to cope with changing conditions and permit more robust tracking. Furthermore, a method for detecting tracking errors controls adaptation to prevent model drift.

2. We present a general spatial and featural sampling strategy for object association that balances short-term with long-term evidence more naturally and reliably than previous methods [29, 4, 72]. We apply it to selectively maintain the relevance of multiple models corresponding to multiple features for small objects moving in cluttered environments and undergoing severe lighting changes and occlusions. We call this framework Adaptive Multi-feature Association, which consists of two components:

   (a) Attribute-based Feature Ranking (AFR) which combines two attribute measures, (a) a measure of discriminability and (b) a measure of independence to other features;

   (b) Multiple Selectively-adaptive Feature Models (MSFM) which involves maintaining a dynamic feature reference of target object appearance. These are updated selectively against current image evidence for ranking features and classifying pixels in each frame.

3. We develop an extension to traditional LBP methods [154, 155] to provide a selective, context-dependent spatial and featural sampling strategy for more efficient and accurate modelling of texton distributions. This approach is able to overcome many limitations of previous methods such as limited spatial support, ad hoc joint and disjoint distributions and resource-limited feature sampling. It may also be integrated with many LBP and LBP-derived methods to enhance them without modification. The framework consists of two components:

(a) A new LBP-type model called Multiscale Selected Local Binary Features (MSLBF);

(b) A novel binary feature selection algorithm called Binary Histogram Intersection Minimisation (BHIM).

1.6 Thesis outline

Chapter 2 provides a review of the current literature. This focuses on sampling strategies that have been employed for object tracking, recognition and feature selection for association. Chapter 3 describes our adaptive spatial sampling strategy for tracking with statistical feature distributions, in particular colour data. On-line adaptation of the statistical model is performed to deal with changes in imaging conditions. Chapter 4 extends this to an adaptive spatial and featural sampling strategy in the form of a framework for tracking by object association over multiple hypotheses that focuses on balancing short-term and long-term evidence to maintain model relevance. The most appropriate features are selected during tracking, with multiple models maintained for the different features and selectively updated. Chapter 5 extends previous work on LBP methods to derive a spatial and featural sampling strategy that more naturally overcomes many of the limitations of the traditional approach. This is demonstrated in the context of recognition tasks. Finally, Chapter 6 offers conclusions and suggests future work.

Chapter 2

Literature Review

The concept of visual sampling is relevant in multiple contexts relating to visual perception. Various tasks, such as tracking and recognition, intrinsically require the ability to perform reliable visual discrimination between various parts of an image. The ability to select the most appropriate features for discrimination greatly supports this goal. In addition, given the fundamentally dynamic nature of real-world environments, relative appearances undergo constant and often drastic change with the result that featural reliability is typically unstable. Consequently, it is necessary to be adaptive in selecting the most appropriate features at any given time as they undergo such fluctuations in reliability. Psychophysical evidence for pre-attentive on-the-fly evaluation of featural discriminability in human vision (and the consequent corresponding attentive salience of targets) is provided in visual search experiments conducted by Theeuwes [218, 219].

The notion of visual sampling is fundamentally rooted in issues such as limited capacity, efficiency, relevance and informational value ([159, 45, 38, 104, 84]). Minimising the effort required in order to maximise the quality and relevance of information received is necessary for vision systems with limited temporal and material resources. Where visual object association is concerned, such objectives may go a long way to being best served by sampling strategies that: (a) incorporate robust predictive mechanisms (see [5]); (b) employ small, descriptive feature sets which are capable of providing sufficient discriminative power for the task at hand (e.g. [93, 102, 76]); and (c) exhibit significant model flexibility and dynamism to cope with changing conditions (e.g. [175, 206]).

In this chapter, previous work is described relating to the contributions made in this thesis with regards to visual sampling for; (a) feature selection for classification; (b) for tracking; and (c) for recognition.

2.1 Feature selection for classification

Many computer vision tasks for which classifiers are to be constructed involve a set of variables which are taken to characterise in some way the underlying differences between objects for which such variables are assigned values. For example, variables relating to pixel hue, saturation and brightness embody the colour appearance difference between a red and a blue object in an image. Depending on the real-world characteristics utilised, a pool of such variables may consist of anything from a few tens to a few hundreds of thousands. Such variables may correspond to a combination of physical *features* that manifest in an image or even just a single feature such as an edge pixel.

To build useful object classifiers, it is important to first select a subset of the most appropriate features to use from the entire pool in order to improve classification performance, especially when training sets are small (Hall [79]). Guyon and Elisseeff [76] and Guyon [74] have previously elucidated several reasons for doing so: (a) it may not be computationally efficient or appropriate to employ all features simultaneously due to problems such as the curse of dimensionality or limited computational and storage resources; (b) too many inappropriate features may pollute predictive power; (c) the most appropriate features will generally demarcate the most relevant differences between classes whereas inappropriate ones may introduce irrelevant categorisations that do not reflect real-world truth, i.e. the best features reflect the underlying generative processes for the sample data; and (d) the identification of the most relevant features may simplify the process of future data collection by better defining where efforts should be directed. More recent benchmarks (Guyon et al. [77, 75]) have suggested that feature selection is not particularly useful for improving classification performance; in particular in cases when selection is performed to avoid overfitting as a result of small datasets. Since such difficulties are mitigated by regularisation techniques, selection becomes more critical for trimming *irrelevant* or redundant features. As such, the problem becomes one of selecting those features that at least minimise any degradation in performance over using the whole set.

In general, a sampling strategy is the process of collecting samples such that they satisfy

some predetermined purpose, such as being representative of a population. Considering the Spatial-Featural Volume (SFV) of an image depicted in Figure 1.8 for vision related tasks; in the same way that pixel samples may be filtered within the 2D spatial domain according to their goodness of fit to a model, they may also be filtered within the third (featural) dimension through a feature selection process operating on feature discriminability. A chosen set of features effectively constrains the slices of the SFV from which sample feature vectors are constructed and subsequently spatially filtered. This process of dynamic feature selection and subsequent spatial filtering can be conceptually viewed as two components of a unified *sampling strategy* which filters samples from the 3D SFV as a whole. The ultimate goal is to form efficient and robust classifiers from compact sets of highly discriminative features with low redundancy. Note that in principle, chosen feature sets should not necessarily be kept fixed after the selection process and in general should be reassessed in line with the contextual dynamics of a vision task.

There are two main classes of feature selection approaches: (1) *filters* [79, 103], which employ feature ranking criteria for selection and operate independently of the induction algorithm to be used (see Figure 2.1); and (2) *wrappers* [93, 108, 21], which "wrap around" specific induction algorithms in evaluating the quality of subsets of features by cross-validation (see Figure 2.2). A third class of algorithm, known as *embedded methods* [76], are similar to wrappers but consist of greater integration between the induction algorithm and the feature searching procedure, with the latter guided by the former. In the case of filters, features are evaluated according to some metric computed on feature distributions (which may be derived from training data or another source). With wrappers, features may be selected on the basis of their actual cross-validation performance, determined by applying the induction algorithm concerned (such as Naïve Bayesian classifiers or Support Vector Machines). We next discuss the ranking of features for filter-based selection algorithms.

Figure 2.1: The "filter" approach to feature subset selection (Kohavi and John [102]). The feature selection algorithm is self-contained and independent of the induction algorithm to be applied.

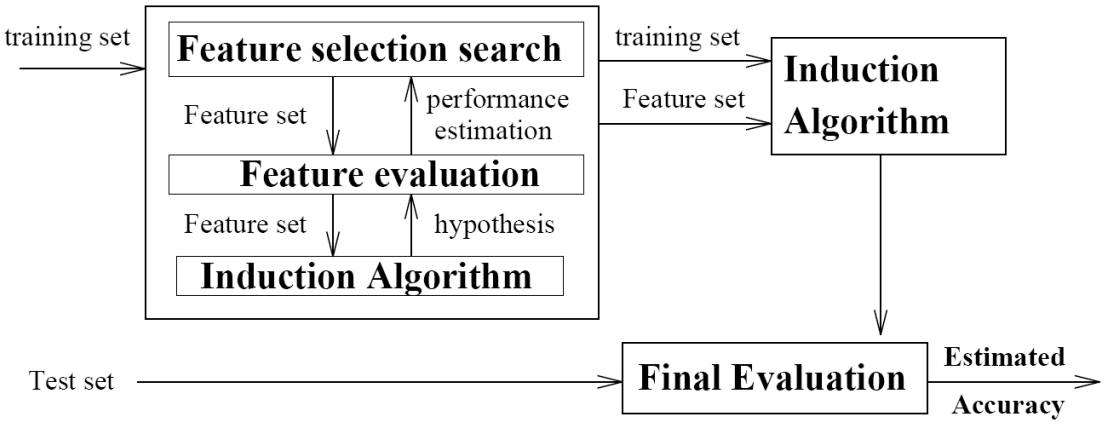


Figure 2.2: The "wrapper" approach to feature subset selection (Kohavi and John [102]). The selection algorithm is "wrapped" around the induction algorithm (classifier) which is used to evaluate the results of adding or subtracting features from the chosen subset. Because of this approach, wrapper methods tend to cater for the biases of individual induction algorithms in choosing the best features; conversely, it also results in a tendency to overfit when training datasets are small.

2.1.1 Ranking features

Feature subset selection is inherently rooted in the notion of the *relevance* of a feature, particularly for filter-based methods. There are several general definitions that have been proposed regarding what constitutes the notion of relevance for a feature and which impinge on a feature selection process. In order to describe these, several notational constructs are introduced. A sample set of K datapoints is referenced as S, with each datapoint $\mathbf{x}_k$, $k = 1..K$ an instance of the vector-valued random variable $\mathbf{X}$ comprising J features X_j, $j = 1..J$. Each feature X_j is drawn from some feature domain $\mathcal{F}_j$. The *instance* space consisting of all possible combinations of feature values is thus given by $\mathcal{F}_1 \times \mathcal{F}_2 \times ... \times \mathcal{F}_J$. A distribution D is defined over the instance space from which the sample set S is assumed to have been drawn. A target function c maps samples to corresponding labels (either deterministically or as a distribution over all possible labels) and encapsulates the idea of a *concept*. A classifier generated using a learning algorithm is denoted as L.

Two important definitions of feature relevance are given in John et al. [93] and Kohavi and John [102] as follows:

Strongly relevant to the sample/distribution: A feature X_j is strongly relevant to the sample

set S if there exist samples A and B in S that differ only in their values of X_j and have different labels (or distributions over labels). More generally, X_j is strongly relevant to c and D if samples A and B have non-zero probability in D, they differ only in their values for X_j and $c(A) \neq c(B)$.

Weakly relevant to the sample/distribution: A feature X_j is weakly relevant to S or c and D if there is a subset of the features which, when removed, makes X_j strongly relevant.

As discussed, there is often the need to reduce the complexity of a classifier (i.e. reduce the resources required for good performance). Embodying the aim of finding compact sets of selected features in order to do this, the notion of relevance as an indication of functional complexity (rather than an evaluation of individual features for selection) is formulated in the following definition (Blum and Langley [15]):

Relevance as a measure of complexity: Given a set of samples S and a set of concepts C, choose the smallest set of relevant features that minimise error over S for one of the concepts in C.

These measures of relevance are independent of any specific form of induction algorithm and features evaluated as relevant are not necessarily useful from the point of view of classification accuracy. A more specific definition of usefulness with respect to a classifier L is given in Caruana and Freitag [21]:

Incremental usefulness: Given a set of previously selected features A, a feature X_j is incrementally useful to L given A if the error rate on S using $\{X_j\} \cup A$ is superior than the error rate using A alone.

This definition has a natural relevance to feature subset selection algorithms, in particular *forward-selection* and *backward-elimination* algorithms (see Section 2.1.2).

In practice, filter methods employ some metric in order to determine how well a given feature X_j predicts the class variable Y. An example of a commonly used criterion for ranking a feature is the Pearson Correlation Coefficient $R(j)$ for feature X_j estimated from K samples (Guyon and

Elisseeff [76]):

$$R(j) = \frac{\sum_{k=1}^{K}(x_{k,j} - \bar{x}_j)(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{K}(x_{k,j} - \bar{x}_j)^2 \sum_{k=1}^{K}(y_k - \bar{y})^2}} \tag{2.1}$$

The Pearson Correlation Coefficient is a normalised measure of linear dependency between two variables. Consequently, a high coefficient between a feature variable X and a class variable Y is taken as an indicator of the utility of X for predicting Y. Largely because of their rigorous theoretical grounding, metrics based on information theory are also commonly employed, such as the mutual information [6] (or information gain [240]) $I(Y;X_j)$ between feature X_j and class variable Y:

$$I(Y;X_j) = H(Y) - H(Y|X_j) = \int_Y \int_{X_j} p(Y,X_j) log \frac{p(Y,X_j)}{p(Y)p(X_j)} dY dX_j \tag{2.2}$$

which is also the Kullback-Leibler divergence $D_{KL}(p(Y,X_j) \| p(Y)p(X_j))$ between the densities $p(Y,X_j)$ and $p(Y)p(X_j)$. Another example of a potential feature ranking metric is the probability of misclassification $\varepsilon(j)$ for feature X_j when employing a Maximum A-Posteriori classifier, where the class $\zeta(\mathbf{x}_j)$ of an instance of the feature $\mathbf{X}_j$ is given by:

$$\zeta(\mathbf{x}_j) = \arg\max_k \{P(C_k|\mathbf{x}_j)\} = \arg\max_k \left\{ \frac{P(\mathbf{x}_j|C_k)P(C_k)}{\sum_l P(\mathbf{x}_j|C_l)P(C_l)} \right\} \tag{2.3}$$

and where the C_ls refer to the classes. $\varepsilon(j)$ for feature X_j is then given by:

$$\varepsilon(j) = \int_{\mathbf{X}_j} \left(P(\mathbf{X}_j) - \max_k \{P(\mathbf{X}_j|C_k)P(C_k)\} \right) d\mathbf{X}_j \tag{2.4}$$

We refer the reader to Forman et al. [58] for a more in-depth comparison of several other measures in the context of text classification tasks. The next section addresses some other crucial issues; that is, those related to the selection of useful *subsets* of features for classification tasks.

2.1.2 Feature subset selection

In the simplest case, features may be ranked by simply applying the ranking criterion for each feature independently and selecting the highest ranked one. This procedure may then be repeated until (for example) the chosen subset reaches some fixed size or a satisfactory level of classification performance is reached. The main drawback of ranking features independently of one another is that the resulting selected subset could possibly contain much redundancy in descriptive or predictive power. For example, replicating the highest-ranked feature and its samples in

a dataset would result in both being ranked highest during the selection process with the corresponding redundancy introduced into the chosen subset. If a fixed-size set is sought, this may potentially prevent the selection of highly complementary features which are individually lower-ranked. Consequently, the process of selecting feature subsets cannot be restricted to simply ranking each variable as a predictor of class on an independently; rather, a more appropriate approach is to rank the predictive power of subsets themselves rather than individual features. The task then becomes one of finding good subsets of features rather than just good features. Such issues have been considered in previous work such as Langley and Sage [108] for selecting features in a forward-selection scheme to improve the performance of naïve Bayesian classifiers. More recently, the concepts of relevancy from the previous section were re-examined by Tsamardinos and Aliferis [224] for *sets of features* rather than individual ones, with the conclusion that relevancy cannot be defined independently of metrics or induction algorithms used. Guyon and Elisseeff [76] showed that the issue of featural redundancy and its relationship to usefulness is further complicated by the realisation that highly-correlated variables may in principle still be complementary to each other. Furthermore, variables which are seemingly irrelevant when considered in isolation may contribute significantly when combined with others (Guyon [74]).

In an ideal world, the subset of features recovered from a larger set for a classification task will contain minimal redundancy and render all other features irrelevant - that is, the chosen subset is *optimal*. This concept is encapsulated in the notion of *Markov blankets* [162]. A Markov blanket is the subset $\mathbf{M}$ taken from a set of features $\mathbf{X}$ such that a random variable Y denoting the class of a set of data is *conditionally independent* of the set difference between $\mathbf{X}$ and $\mathbf{M}$ given $\mathbf{M}$:

$$Y \perp\!\!\!\perp (\mathbf{X} \backslash \mathbf{M}) | \mathbf{M} \tag{2.5}$$

The Markov blanket $\mathbf{M}$ then defines the subset that excludes those features that do not contribute any information for determining Y. It has been shown to potentially constitute the set of *strongly relevant* (see Section 2.1.1) features (Tsamardinos et al. [225]). Another perspective may be provided by the notion that the optimal subset $\hat{\mathbf{M}}$ minimises the conditional entropy $H(Y|\hat{\mathbf{M}})$ between Y and features in $\hat{\mathbf{M}}$ taken jointly (Fleuret [55]):

$$\hat{\mathbf{M}} = \arg\max_{\mathbf{M}} \{H(Y|\mathbf{M}) | \mathbf{M} \in 2^{\mathbf{X}}\} \tag{2.6}$$

where $\hat{\mathbf{M}} = \{X_{b_1}, X_{b_2}, ..., X_{b_K}\}$ and b_k denotes the index of the k'th feature selected from $\mathbf{X}$.

The problem of selecting optimal feature subsets suffers from the same drawback as many other pursuits involving the optimisation of complex multimodal functions; namely the computational difficulty with finding a global optimum. Corresponding optimisation algorithms tend to address a trade-off; that of minimising computational effort for the cost of finding only local minima. Inherent to this situation is the sensitivity of such algorithms to initialisation, which in general strongly influence the local minimum found during a search. For a feature set size of J the number of possible combinations and subset sizes K are:

$$K = \sum_{k=1}^{J} \frac{J!}{k!(J-k)!} \tag{2.7}$$

which leaves an exhaustive search for optimal subset sizes and corresponding selected features computationally prohibitive. Figure 2.3 illustrates the tree of possibilities for a feature selection task involving a pool of four features.

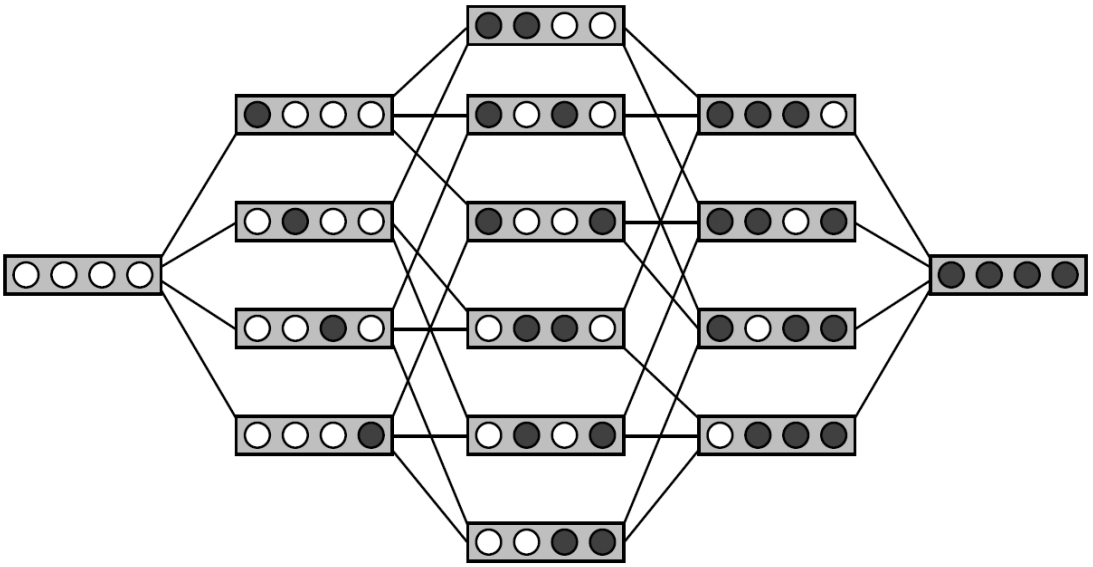


Figure 2.3: The computational intractability of exhaustive feature subset selection (Blum and Langley [15]). Selecting subsets from a pool of four features requires first choosing the first feature (left) and then subsequently adding features to complement the one(s) already chosen (moving rightwards). As the pool increases, the set of possible combinations at each stage become prohibitively large. Consequently, in practice most algorithms can only traverse suboptimal subpaths through the tree which are highly dependent on the initial starting point and the criteria used to evaluate features.

Many feature subset selection algorithms involve a greedy *forward selection* or *backward elimination* procedure for selecting subsets of features from a feature pool. The former involves starting with an empty set and incrementally choosing features from the pool to add to this set until some criterion reflecting the quality of the chosen set is met, such as a negligible increase in classification performance. The latter, on the other hand, involves starting with the entire pool and incrementally removing features one-by-one. This may be done, for example, by choosing features which, when removed, have minimal impact on classification performance, denoting their "irrelevance" to the class variable. Such algorithms are greedy since once a feature has been selected, either for addition or removal, the decision is not revisited at a later date. Some algorithms take a hybrid non-greedy approach consisting of steps that add and subtract features in order to find the ideal Markov blanket (see e.g. Tsamardinos et al. [225] for an overview). While features may be chosen on the basis of their impact conditional on the current chosen set, this does not guarantee optimality since a single feature in isolation may be either a good predictor of the class variable or a seemingly bad one but change in such a relationship when taken in conjunction with others [76, 74]. Consequently, such algorithms are inherently suboptimal and may be viewed as finding the equivalent of a local extrema in the space relating all feature subsets with classification performance.

2.1.3 Filters vs wrappers

There has long been debate regarding the relative merits of filter and wrapper methods for feature selection. As previously described, filter methods [79, 103] employ metrics of "goodness" to evaluate features as a pre-processing step independently of any particular induction algorithm. As such, the biases of the induction algorithm play no role in which features get selected, unlike wrapper methods [93, 108, 21] which, due to the cross-validation component of the approach, tend to find subsets more tuned to the particular characteristics of the form of classifier. This fact has been used to support the use of wrappers over filters [101]. However, wrappers tend to be computationally expensive due to the need to test performance for each feature candidate at each stage of adding or subtracting a feature, whereas filters are more scalable to the size of training sets. Moreover, wrapper methods appear to overfit on small training sets [79]. On the other hand, they can be more widely applicable since the induction algorithm to be used can be treated as a "black box", with the same wrapper algorithm usable for a multitude of classifiers. Hybrid methods which borrow characteristics from both filters and wrappers have also been proposed

(e.g. Das [42], Sebban and Nock [191], Ni and Li [150]). Das [42] offers a discussion of the relative merits of filter and wrappers and argues with experimental justification that, for real-world datasets, features that enable good performance with one kind of classifier should provide similarly good performance on another, even if the selected feature set is suboptimal for that method. A hybrid algorithm was proposed which uses boosting and combines some features of wrappers into a filter method without increasing computational cost. Guyon and Elisseeff [76] also discuss *embedded* methods which perform selection during the process of training a predictor as opposed to the more expensive procedure of training predictors for each subset of features picked out by a wrapper.

Boosting is a procedure for combining a set of "weak" classifiers which individually may not perform much better than random but when taken in a weighted combination (or *ensemble* constitute a "strong" classifier capable of significantly greater performance. AdaBoost (Adaptive Boosting) is a widely used and built-upon algorithm for boosting [61] and is commonly used for feature selection, such as for iris recognition [24], acoustic event detection [255], face detection [228, 236], face recognition [197] and facial expression recognition [198, 196, 195]. The algorithm is based upon the idea of training classifiers on the basis of weights assigned to $\mathrm{p}(x_j, y_j)$ pair samples in a training set. Initially, weights are uniform across all samples. The first classifier is trained on the equally-weighted set and added to the ensemble. The sample set is then reweighted such that incorrectly classified samples are assigned higher weights. A new classifier is then trained with the new weights and added to the ensemble. The cycle continues until a certain number of classifiers have been selected (see Figure 2.1). If the training data contain many features per sample, this scheme may be employed as a feature selection process. In this case, the AdaBoost classifier training step is replaced by an evaluation of each of the features in the set (given some induction method) on the currently weighted training samples, with the best predictor chosen at each stage. As such, AdaBoost is a wrapper method since samples are reweighted according to the performance of the induction algorithm concerned. However, selected features may often be highly correlated with one another.

A good recent example of a filter which tackles the problem of featural redundancy is the Conditional Mutual Information Maximisation (CMIM) algorithm [55], based on the concept of conditional mutual information:

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} \left[h_t(x_i) \neq y_i \right].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

Algorithm 2.1: The AdaBoost algorithm for binary classification tasks (Freund and Schapire [62]). The procedure involves training an ensemble of classifiers in sequence, each one on training data re-weighted according to the performance of those classifiers already trained. Consequently, "hard" examples are given greater focus of attention with the result that trained "weak" classifiers taken together perform more strongly. This method may also be used as a wrapper for feature selection (refer to text).

$$I(U;V|W) = H(U|W) - H(U|W,V) = \int_W \int_V \int_U p(U,V,W) log \left\{ \frac{p(W)p(U,V,W)}{p(U,W)p(V,W)} \right\} \quad (2.8)$$

where U, V and W are random variables and $I(U;V|W)$ is the mutual information between U and V given W.

CMIM is a computationally efficient greedy feed-forward selection algorithm which selects candidate features X_j which maximise their mutual information with the class variable Y *conditional on the features already picked* $\{X_{b_1}, ...X_{b_k}\}$. However, since doing this would require the prohibitive step of estimating large joint densities (i.e. $I(Y;X_j|X_{b_1}, ..., X_{b_k})$), a trade-off is made which involves selecting the feature X_j which maximises its mutual information with Y conditional on the feature from the set already picked that *minimises* its mutual information with X_j:

$$b_1 = \arg\max_j I(Y;X_j), \quad b_{k+1} = \arg\max_j \left\{ \min_{l \leq k} \{ I(Y;X_j|X_{b_l}) \} \right\} \tag{2.9}$$

This forms an efficient, suboptimal trade-off which at most requires estimating joint distributions of triplets of variables. Moreover, efficiency can be further improved when working with binary data as the joint distributions can be estimated by simple summation operations. The algorithm has been shown to improve performance over other methods such as AdaBoost.

In Chapter 4, we embody the notions of reducing featural redundancy in a novel approach to selecting features on-the-fly for object tracking under very challenging conditions. We combine a measure of featural discriminability with Canonical Correlation Analysis (CCA) [86] as a relatively computationally-inexpensive approximation to evaluating featural redundancy in order to improve the discriminative power of chosen feature sets during tracking. Experiments demonstrate significant improvement over previous feature-selection based tracking techniques. In Chapter 5, we describe a novel, computationally efficient filter for binary feature selection called Binary Histogram Intersection Minimisation which is experimentally shown to find stronger subsets of low-redundancy binary features than either CMIM or AdaBoost. Experiments demonstrate its effectiveness on both synthetic datasets as well as Local Binary Pattern (LBP) feature selection tasks for texture and face discrimination (see Section 2.3).

2.2 Sampling for tracking

The need for reliable methods for tracking objects as they change state over time (for example, estimating an object's position at each time step as it moves through a visual field) has driven much research in the past few decades, largely due to military requirements (e.g. [200, 190, 199, 181, 60]). Tracking in general involves deriving measurements from sensory data reflecting some state(s) of objects of interest and updating some corresponding state representation for these objects over time. As such, measurements at different time steps must be associated with the appropriate dynamic physical entity (motion correspondence). However, it is recognised that sensory measurements are generally noisy, uncertain and often incomplete [200], necessitating methods for handling such inconveniences by incorporating models of noise in updating state estimates. Furthermore, tracking the states of multiple objects simultaneously often necessitates a disambiguation of possible correspondence assignments to establish a correct one-to-one mapping and avoid the erroneous convergence of multiple separate pieces of evidence onto a

single track. Consequently, tracking involves combining measurements with predictions at each time step. *Data association* methods and algorithms [5] have been developed for this purpose, employing rigorous mathematical procedures for the optimal association of measurements with *tracks* (the history of object state(s) over time) and their subsequent updates. The predictive component of data association essentially forms part of a *sampling strategy*, defining hypotheses for the likely state(s) of objects. Given such hypotheses, the sensory data from which measurements are derived are filtered accordingly in computing final state estimates. They can also help to ensure greater efficiency of effort, for example in providing a region-of-attention in visual tracking tasks.

The types of information sampled from sensory data for tracking purposes can be hugely varied. Military applications most frequently employ radar or infrared features since these are the most robust for specific scenarios, such as tracking aircraft beyond visual range or missiles designed to home in on the heat signatures of moving enemy vehicles. In these circumstances, sensory data consists of the detected spatial locations of targets and the data association problem becomes one of matching detections to the correct tracks as well as updating those tracks over time. Such data may not reflect the *identity* of the object responsible for it; consequently, since each track constitutes the state history of a single object over time, a data association process must inherently perform *object association* to match specific detections to specific tracks. Where sensory data may provide information regarding object identity, object association may be performed independently of track updates. For example, in vision-based tracking tasks such as surveillance or human-computer interaction, the samples employed tend to be features derived either directly from the image, such as raw chromatic data (e.g. [234, 176]), or from some transformation of the image, such as geometric features (e.g. [154, 121, 41]). Objects may then be modelled according to their characteristics within the context of such features and object association performed on the basis of correlations between image evidence and object models, with measurements derived from strongly associated image data. The tracking process then involves the search for image evidence that correlates well with what is expected by the model, along with any predictive element provided by data association techniques to aid performance. While objects may be modelled (and detected) directly according to their visual appearance, much work also involves the converse (but largely equivalent) task, that is modelling *background*, with foreground objects implicitly isolated by virtue of not matching background expectations (e.g.

[206, 207, 185]). There are also methods for performing classification on image samples during tracking to determine whether they belong to the object or to the background, including recent sophisticated feature-selection based approaches [30, 4, 72].

In this section, we first describe some key work in modelling foreground object appearance. Secondly, the converse task of modelling of background appearance statistics for isolating foreground objects is discussed with some recent methods. Finally, we cover recent, more sophisticated tracking techniques that look to more rigorously distinguish between object and background by performing foreground/background classification, including methods that dynamically select the most appropriate features to use for performing the classification step.

2.2.1 Modelling foreground

Many methods for tracking involve modelling the expected appearance characteristics of an object and estimating object state, such as position or pose, by collecting image samples and evaluating them in some way. These evaluations may then be used to update the relevant quantities being monitored. Such evaluations may take the form of probabilities or functions reflecting real-world constraints for extremisation. Data association methods (see Bar-Shalom and Fortmann [5]) form the complementary function of coordinating these potentially noisy samples with prior expectations of object state, with the added practical benefit of providing a sampling strategy for the measurement process, such as a region of focus-of-attention to reduce computational overheads.

Object appearance may be captured in a variety of ways, for example by modelling photometric statistics such as colour or geometric quantities such as contours (e.g. [243, 10]). The latter generally consist of features that capture some physical set of geometric properties of the object(s) concerned, such as shape information or silhouettes. These tend to be computationally expensive as well as sensitive to various real-world difficulties such as occlusions and viewpoint changes. Some examples of previous work employing geometric properties include articulated body tracking [180, 184, 145, 164, 66] which model human bodies with connected-component representations and estimate pose during tracking, silhouette-based person segmentation [7, 8, 11, 47, 167] which find silhouette regions corresponding to human bodies, snakes [96] which are flexible contour models that adapt to find physical edges in images, active contours [40, 13] which characterise and constrain the expected shapes of objects for localisation during tracking and active shape models [37, 36, 35] which comprise multiple adaptive shape objects

that search for physical topologies in images within a constrained range of geometric arrangements (such as eyes and lips for a face). These geometry-based methods depend on *edge* features in an image which can be recovered from raw pixel values by applying edge detection methods [20, 16]. Objects may also be characterised by *texture* appearance (e.g. [80, 133, 73, 154]), which essentially amount to arrangements of local edge features and as such fall into the geometric category. Geometric features tend to be quite robust against difficulties such as illumination changes since in most cases image edges may be consistently recovered regardless of the specific brightnesses or chromaticities of the physical objects that depict them. However, photometric features such as colour and brightness are often favoured over geometric ones because of their greater robustness in many other real-world situations such as partial occlusions, object non-rigidity, resolution changes and viewpoint and scale changes.

Colour is commonly used for characterising photometric object appearance. It can be very useful when colours of an object of interest are distinct against background and distractors; in such situations, colour alone can often suffice as a robust, computationally cheap and efficient discriminator. Although RGB is a standard representation employed, several different colour spaces exist as alternatives to RGB with different characteristics, such as L*a*b* (e.g. [18]), YUV (e.g. [31]) and Hue-Saturation-Value (HSV) (e.g. [100, 188, 203]). Zarit et al. [245] have provided a comparison of several colour spaces for representing skin colours. As well as object tracking, colour has been used for segmentation [201], recognition tasks [134, 212] and detection and tracking specifically for human faces by modelling skin colours as distinctive features (e.g. [100, 188]). Often, object colour statistics are modelled as probability distributions to facilitate probabilistic induction schemes. Histograms are commonly employed as non-parametric approximations of such distributions. In a landmark work, Swain and Ballard [212] introduced the use of colour histograms as a fast method for representing object appearance for real-time recognition, noting the aforementioned benefits over geometric features. Their work covered three main areas: (1) The histogram-based representation scheme for characterising object appearance; (2) Two algorithms for matching histogram models, namely *Histogram Intersection* for matching the histograms of two objects and *Incremental Intersection* for efficient indexing into large databases; and (3) *Histogram Backprojection* for finding regions of an image corresponding to objects. The general drawbacks of histogram-based approaches to approximating distributions include the need to choose a priori the partitioning of the input space (number and size of bins),

which is an instance of the well-known model selection problem. Also, small numbers of data for initial training may result in sparse histograms which do not provide an adequate model. Although the original work [212] focused on object recognition, tracking using such a model (or similar representation) may be performed by searching for regions of expected shape and orientation which exhibit a high affinity (e.g. probability) with the model. For example, kernel-based tracking [33] or simple mean-shift on raw pixel evaluations [32], which at each frame efficiently search for local maxima in the search space (e.g. position or size) relative to the previous frame, may be applied for updating object state estimates.

One of the reasons for the robustness of colour against geometric deformations is that most colour models do not incorporate spatial configuration. Simply modelling the statistics of object colours ensures that under most geometric transformations the distribution is largely stable. This can also greatly simplifies the step of performing correspondence between frames. On the other hand, ignoring a major characteristic such as geometric topology can be counterproductive since they provide another constraint; under real-world situations such as moving cameras and other distractors, the distinctiveness of colour alone will change from time to time. A popular technique that essentially combines photometric quantities with geometric elements for modelling and tracking deformable objects is *template tracking*, where an object of interest is characterised by a template comprising a region of pixels. This involves mapping an image region consisting of pixels $\mathbf{x}$ from an image for frame t I_t to the next frame I_{t+1} by optimising the parameters $\mathbf{p}$ for a geometric warp function $\mathbf{W}(\mathbf{x};\mathbf{p})$ of some kind, which transforms the position and/or shape of the template to best fit a similar region in frame I_{t+1}. As such, the technique intrinsically captures both the colour (photometric) appearance of objects as well as their spatial (geometric) layout, with tracking involving the search for plausible deformations of the template between frames. More specifically, given a template $\mathrm{T}(\mathbf{x})$ generated from the pixels of the object in frame 0, $I_0(\mathbf{x})$ (Lucas and Kanade [123]):

$$\mathbf{p}_t = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}\in T} \left[I_t(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})\right]^2 \tag{2.10}$$

Generalising the method to allow templates to be updated and indexed by frame number T_t, the naïve approach to template update is to replace the template at frame t by the warped template from frame $t-1$:

$$T_{t+1}(\mathbf{x}) = I_t(\mathbf{W}(\mathbf{x};\mathbf{p})), \ \ \forall t \geq 1 \tag{2.11}$$

Model drift in this context is introduced due to small errors in the optimisation process resulting in non-object pixels being introduced into the template T_{t+1}. Matthews et al. [135] formulated a method for counteracting this model drift by embracing a similar concept to that of Collins et al. [29] (see Section 2.2.3) and using the template from the first frame T_1 as an "anchor" for warp parameter optimisation. This is done by a two-stage optimisation process using gradient descent; first estimating $\mathbf{p}_t$ using T_t and $\mathbf{p}_{t-1}$ as the starting point for gradient descent and thereafter optimising again using T_1 and $\mathbf{p}_t$ to derive the final estimate $\mathbf{p}_t^*$:

$$\mathbf{p}_t = gd \min_{\mathbf{p}=\mathbf{p}_{t-1}} \sum_{\mathbf{x}\in T_t} [I_t(\mathbf{W}(\mathbf{x};\mathbf{p})) - T_t(\mathbf{x})]^2 \tag{2.12}$$

$$\mathbf{p}_t^* = gd \min_{\mathbf{p}=\mathbf{p}_t} \sum_{\mathbf{x}\in T_1} [I_t(\mathbf{W}(\mathbf{x};\mathbf{p})) - T_1(\mathbf{x})]^2 \tag{2.13}$$

where *gd* indicates a gradient descent minimisation of the error function. Template update is then performed using the warping with $\mathbf{p}_t^*$ rather than $\mathbf{p}_t$:

$$T_{t+1}(\mathbf{x}) = I_t(\mathbf{W}(\mathbf{x};\mathbf{p}_t^*)) \tag{2.14}$$

with the caveat that if the difference between the first and second stage optimised parameters $||\mathbf{p}_t^* - \mathbf{p}_t|| > \varepsilon$ where ε is a small threshold then an error is assumed and the template is left unchanged $T_{t+1}(\mathbf{x}) = T_t(\mathbf{x})$.

Nguyen and Smeulders [149] proposed a template update method which incorporates temporal smoothing to counteract the effects of occlusions or sudden lighting changes on template updates. Noting that temporal filtering based approaches such as Kalman filters and their extensions tend to be useful for smoothing motion trajectories rather than helping to localise objects in subsequent frames, their approach involves maintaining an object appearance model comprising *photometric feature vectors* for pixels from the target region and applying Kalman filters to "track" individual feature vectors from the template as they change over time. Feature vectors in subsequent frames are estimated within a Kalman framework through a combination of an observation model approximated by a Gaussian and a prediction model. Outliers are assumed to be caused by occlusion or sudden lighting changes while gradual changes are accommodated.

While the method is capable of handling complete occlusion by suspending feature vector tracking when measurements deviate significantly from the model, it requires an upper limit to be placed on the number of successive frames in which an object can be assumed to be occluded. It can also adapt undesirably to non-object regions during long-term partial occlusions.

Whichever methods are used for performing correspondence between frames and whichever features are used, whether photometric or geometric, matching may be viewed as a process of filtering samples. Image themselves are pools of samples of a real-world imaging process and tracking essentially seeks to focus on a subset of samples at each frame by estimating object position and even also size and orientation. For example, computing probabilities for pixels with respect to a probabilistic model can be viewed as assigning weights for a filtering process, with methods such as mean-shift [32] subsequently performing the final filtering step by looking for local maxima in the weight image. Optimising error functions for active contour tracking effectively seeks to determine those edge pixels samples which fall within the constraints of acceptability for expected object shape. For template tracking where error minimisation is performed, the samples are effectively filtered during the warp parameter estimation process with the final sample set corresponding to the local optimum parameterisation. In all cases, the state in frame t provides an implicit sampling strategy by imposing limits on acceptable state estimates for frame $t+1$ (e.g. image probabilities for frame $t+1$ only need be computed within a certain proximity of the position and size estimated by mean-shift in frame t).

As previously discussed, photometric quantities such as brightness and colour are inherently unstable under changing lighting conditions. Consequently, the sampling strategy employed for object association on the basis of such quantities needs to be flexible in order to cope with most real-world situations, such as by dynamically adapting the model of object appearance to reflect such changes. As a consequence of adapting object models, the problem of model drift caused by the incorporation of non-object samples into a model becomes an issue. In Chapter 3 we develop a framework for modelling the statistics of complex multi-coloured objects using Gaussian mixture models and demonstrate the use of these models for real-time object tracking using extremely modest hardware. Further, the framework incorporates a mechanism to adapt the model on-line to keep track of appearance changes due to lighting. A method for automatically detecting tracking errors is employed in order to establish selective adaptation; that is, when the tracker is deemed to have failed adaptation is suspended to reduce the chance of model drift.

These two components together form a flexible sampling strategy that seeks to maintain the effectiveness of pixel sample filtering in each frame. Experiments demonstrate its effectiveness for tracking objects under severe lighting changes and preventing model failure by selectively suspending adaptation. In Chapter 4 we develop a novel framework that builds upon previous feature-selection based tracking techniques that attempt to avoid model drift by employing static object models. Our framework maintains models for each feature type and performs selective pixel sampling and model adaptation to enable feature-selection based tracking that can deal with longer-term object appearance changes under extremely difficult conditions while reducing the risk of model drift. We demonstrate its effectiveness in comparison with established methods in several experiments depicting the tracking of target objects at a distance from the camera in highly cluttered low-resolution environments whilst undergoing severe lighting changes and significant temporary occlusions.

2.2.2 Modelling background

An alternative, but complementary, approach to modelling the appearance of an object in a scene for a tracking task is to model the *background*. In doing so, it becomes possible to remove it in each frame, leaving only foreground objects of interest which may then be further examined for estimating object state. In the simplest case, a single template or reference image of the empty background for the scene of a tracking task may be subtracted from each image. However, such an approach is severely limited in several ways. While the term "background" may intuitively be assumed to refer to "static" elements of the scene, its meaning is inherently flexible and context dependent. For the purposes of vision tasks it may effectively be taken to refer to aspects of the scene which are *not* of interest. This may include both static and dynamic elements, such as buildings or animals. Furthermore, static background objects may become dynamic, such as a parked car being driven away or a door being opened. Moreover, outdoor scenes are subject to other difficulties, notably changes in illumination due to changing sun positions and cloud cover with resulting shadowing, as well as wind-induced motion such as swaying trees. Consequently, the modelling of a background is *fundamentally* a statistical endeavour which needs to be flexible enough to adapt to changing conditions.

Background models may in general utilise three different categories of features; *spatial*, *spectral* or *temporal*. Spatial features employ local structural information such as gradients or textures. Spectral features make use of chromatic (colour) or brightness information and are the

most commonly employed feature [206, 81, 51]. Temporal features use estimates of motion such as those yielded by frame differencing or optic flow algorithms. Some work combines spectral information with spatial to add robustness against illumination changes [158, 91] and others have employed temporal features to characterise dynamic pixels corresponding to nonstationary objects [222, 110, 233].

Wren et al. [234] describe a system known as Pfinder which is used for the tracking and segmentation of people as well as the interpretation of their actions. Pfinder combines colour and shape features within a Maximum A-Posteriori (MAP) framework by maintaining a blob representation based on work by Pentland et al. [163, 97]. Blobs are formed by assigning feature vectors to each pixel comprising appearance (i.e. colour) along with their spatial coordinates and clustering, resulting in a collection of spectrally and spatially coherent image regions (blobs). This collection forms a representation of a human body where individual blobs related to specific physical parts of the person being tracked (see Figure 2.4). Their framework incorporates a statistical method of modelling the scene, which they use in classifying pixels in each frame as scene or one of the blobs in the body model. This model involves estimating the parameters of a single Gaussian (mean μ and covariance Σ) for the colour of each pixel of the image as part of an initial learning process. During tracking, the likelihood of each pixel belonging to its corresponding model is computed. Deviations beyond a certain degree are taken to be caused by the person being tracked occupying that point in the image and the pixel taken to be foreground for assigning to a blob. This simple statistical approach has some drawbacks, namely the assumption that the scene, once learned, remains static. The use of a single Gaussian supports this assumption since a single mode colour distribution is assumed for each pixel, useful only for modelling small-scale fluctuations, for example due to imaging noise. Because of real-world conditions such as wind-induced motion or lighting changes, outdoor scenes are more likely to exhibit multiple colours at individual points [64], necessitating the ability to model multimodal colour distributions for each pixel.

Stauffer and Grimson [206, 207] employ adaptive Gaussian mixture models to represent multiple colour distributions for individual pixels in a scene with a static camera. Each Gaussian effectively corresponds to a single expected colour. When processing a frame, the colour of each pixel is compared to each associated Gaussian. If the colour lies within 2 standard deviations of a distribution, that distribution is updated with the new colour, otherwise the pixel is classified

Figure 2.4: The Pfinder system developed by Wren et al. [234]. The left image is the input image. The middle image depicts the grouped chromatically coherent regions for blob model generation. The right image illustrates the resulting blob model, with each blob relating to a single coherent region as generated by the segmentation step.

as part of a moving foreground object. In other words, foreground pixels can be detected as colours least likely to have been produced by background processes as modelled by the mixtures. This method is able to model changes over time in outdoor scenes caused by slight movements (such as swaying trees) and deal with lighting changes including shadows moving over time. Foreground pixels may be combined using a connected components algorithm. In this fashion, whole foreground regions can be processed for tracking using data association techniques (see Figure 2.5).

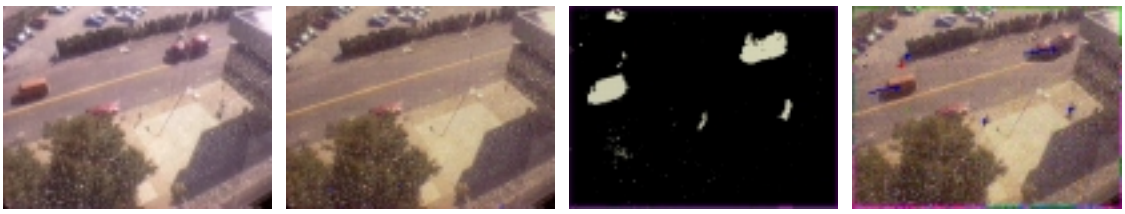


Figure 2.5: An example of tracking using adaptive Gaussian mixtures to dynamically model per-pixel background colour (Stauffer and Grimson [206]). During tracking, pixels sufficiently deviated from their background models are considered to be foreground. The images show (from left to right); first the input image, followed by the image constructed from the means of the most probable Gaussians for each pixel. Third, a binary background/foreground image and finally the result of tracking on the foreground binary image.

There have been developments of the statistical model approach with surveillance applications in mind, such as by Cavallaro and Ebrahimi [22] (see Figure 2.6), aimed at the recovery of multiple object boundaries over long periods of time and in the presence of camera noise. The

recovery of object boundaries when separating foreground and background may be useful, for example, in order to maximise the utility of visual features such as shape and gait for recognition and motion analysis. The background is learned and updated in an on-line fashion to counteract changes due to slow lighting fluctuations and used as a reference frame for image subtraction and object detection. Camera noise can be explicitly modelled in order to reduce errors in estimating object boundaries. However, many such techniques assume that the camera being employed is static so that the topological structure of the background as imaged by the camera remains as geometrically stationary as possible.

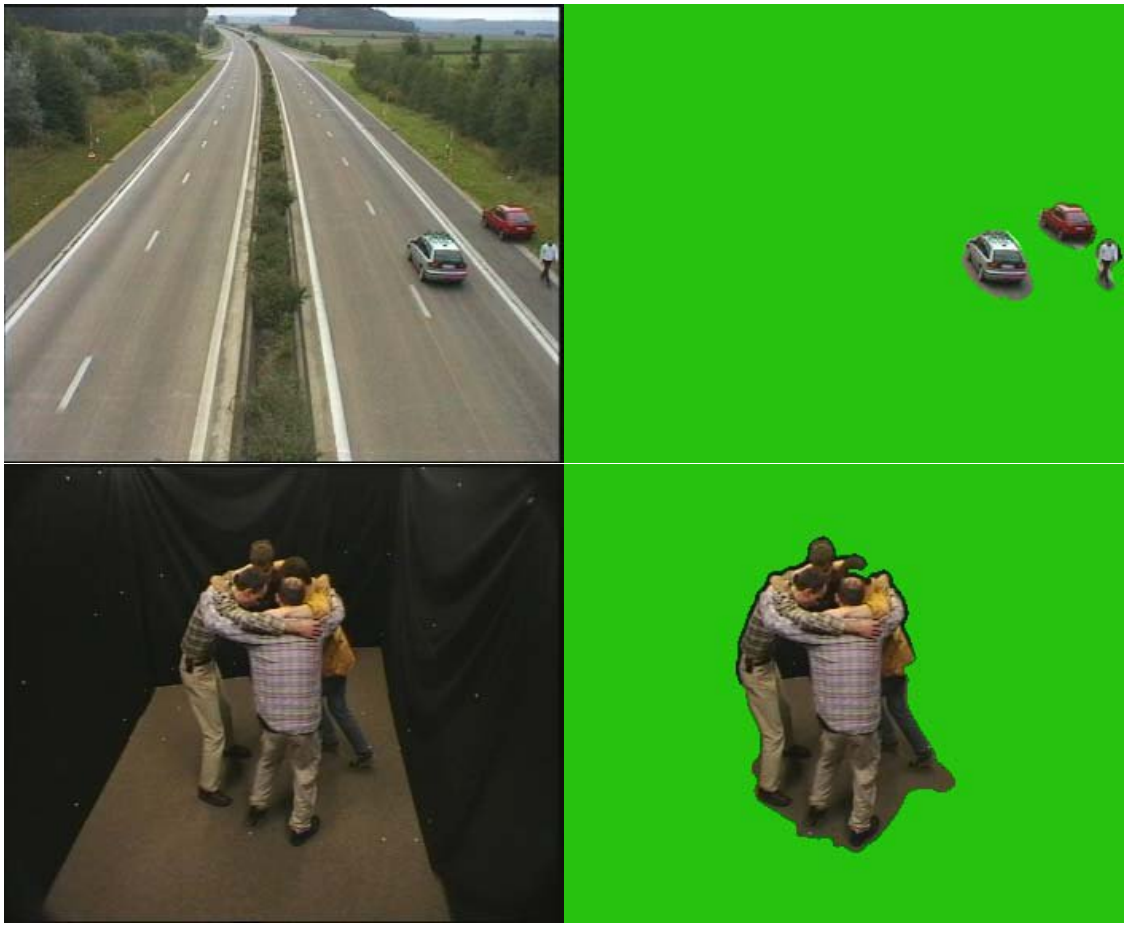


Figure 2.6: Example of foreground extraction with an adaptive background model (Cavallaro and Ebrahimi [22]). The left column shows input images and the right column corresponding extracted regions.

An effort is made to systematically employ spatial, spectral and temporal features together within a Bayesian framework for foreground detection by Li et al. [111], wherein background pixels are represented by *principal features*, those which are the most significant and frequent.

A Bayes decision rule is used for classification based on the statistics of these principal features and a learning method proposed for the adaptation of the background model to both gradual and sudden changes. Figure 2.7 illustrates foreground extraction from a difficult scene containing specular highlights and a moving escalator. However, limitations of this technique include the inability to distinguish static foreground objects from the background with the consequence that moving regions which become static for lengthy periods of time can become absorbed into the background (although in some cases this may be desirable). Additionally, it can be sensitive to crowded regions.

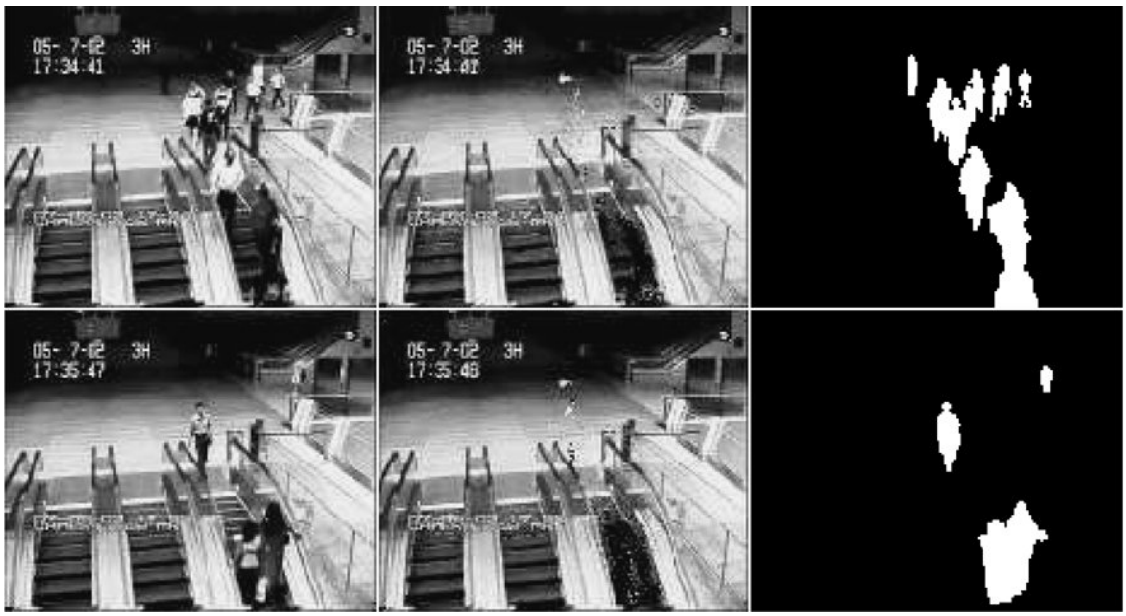


Figure 2.7: Example of foreground extraction using spatial, spectral and temporal features in a unified Bayesian framework (Li et al. [111]). The scene contains difficult lighting and dynamic scene components (an escalator).

Russell and Gong [185] built upon work by Cohen [28] by taking a combinatorial optimisation approach to estimating background images. Given a *block* of sample frames taken from a video sequence, a minimum graph-cut algorithm [57] is applied to minimise a labelling cost function which dictates spatial and temporal consistency at each pixel location. In doing so, each pixel of the background may be estimated from different frames of the block, with highly successful results. A major benefit of such an approach is that one does not require an examination of the empty scene; rather, the empty scene may be recovered from a collection of sample image frames as long as the background colour for each pixel is adequately represented in the sample set.

Other methods for background modelling and subtraction include the use of spatial information as well as colour statistics (e.g. [51, 112, 222]), HMMs for scene event-based pixel classification [182, 208], subspace analysis [156] and the modelling of images as auto-regressive moving average processes [147, 253]. A more detailed survey of such methods is provided in [242].

To summarise, these methods can serve to act as a complementary approach to modelling object appearance by modelling background appearance at each spatial location and removing pixels considered to be the background, leaving pixels deemed to belong to foreground objects. Although some methods are able to deal with changes over time in the background caused by changes in lighting or low-level motion such as swaying leaves, the inherent spatial encoding of such models require the camera to remain static. Consequently, such background removal methods are unable to deal with moving cameras since slight changes in camera pose cause global changes in projected scene geometry. In Chapter 3 we demonstrate a framework that allows for the modelling of background colours using Gaussian mixture models along with a Bayesian tracking algorithm that employs these models in conjunction with foreground object models for real-time tracking with a moving camera. Such models can embody complex distributions of multiple colours without regard to their spatial organisation. Furthermore, they are amenable to the use of computationally inexpensive adaptive mechanisms to update the background representation over time to cope with changing geometry and lighting conditions.

2.2.3 Foreground-background classification

A generic tracking methodology must be capable of dealing with a wide range of conditions. One difficulty is in maintaining the tracking of an object under arbitrary camera motion. Moving cameras are inevitably accompanied by drastically changing backgrounds; moreover, other real-world problems such as lighting changes and occlusion also mean that the *foreground itself* is likely to exhibit such drastic changes at the same time. More specifically, the most appropriate features to use for discriminating between background and foreground are *intrinsically unstable* and liable to change constantly over time. For example, a person wearing a red coat moving amongst people all wearing black can be easily tracked simply by using colour; however, if everyone else wears red a geometric feature such as body shape may perform better. From a computational point of view, modelling the foreground or background alone may often provide good results but in many situations both foreground and background will share characteristics

which warrant the use of additional or alternative features. This inherently dynamic and often ambiguous nature of both foreground *and* background forms a challenging hurdle for tracking methods to overcome and which necessitates a flexible, adaptive approach for models to be kept up-to-date and maintain contextual relevance over time.

To deal with such situations, recent tracking methods employ a binary classification approach to label pixels as object or background, rather than just modelling one or the other. As such, background appearance may be traded off with foreground characteristics. Comaniciu et al. [33] achieve this by reducing the weight of foreground colours that are also represented in the background and applying mean-shift [32]. Collins et al. [29] developed a framework for the selection of the most discriminative out of a pool of features during a tracking task which are then used to classify pixels. The tracker is able to switch between feature spaces on-the-fly in order to maintain discriminative potency and adequately separate the object from the background. Their experiments were conducted using a feature pool of 49 different unique linear combinations of *RGB* channels $w_1R + w_2G + w_3B, w_* \in \{-2,-1,0,1,2\}$ (see Figure 2.9). Those linear combinations which are scaled versions of each other $(w_1', w_2', w_3') = k(w_1, w_2, w_3)$ were trimmed as redundant (for example, $-2R+2G-2B$ is removed since it is $-R+G-B$ multiplied by 2).

For this framework, in each frame t pixels are sampled within a centre-surround arrangement corresponding to a fitted bounding box (see Figure 2.8). For each feature X_j in the pool, feature values $\mathbf{x}_j^k \in \mathcal{F}_j$ are extracted from the corresponding foreground and background regions and normalised distributions $p(X_j|F_t)$ and $p(X_j|B_t)$ respectively formed, where $\mathcal{F}_j$ is the set of all possible values for feature X_j. The features are then ranked according to the variance ratio $VR_{j,t}$ (see Figure 2.10):

$$VR_{j,t} = \frac{var\left\{j,t,\frac{1}{2}\left[p(\cdot|F_t)+p(\cdot|B_t)\right]\right\}}{var\left\{j,t,p(\cdot|F_t)\right\}+var\left\{j,t,p(\cdot|B_t)\right\}} \tag{2.15}$$

where

$$var\left\{j,t,\Psi(\cdot)\right\} = \left\{\sum_{\mathbf{x}\in\mathcal{F}_j}\Psi(\mathbf{x})\log^2\left(\frac{p(\mathbf{x}|F_t)}{p(\mathbf{x}|B_t)}\right)\right\} - \left\{\sum_{\mathbf{x}\in\mathcal{F}_j}\Psi(\mathbf{x})\log\left(\frac{p(\mathbf{x}|F_t)}{p(\mathbf{x}|B_t)}\right)\right\}^2 \tag{2.16}$$

During tracking, pixels are weighted according to the N highest ranked features from the previous frame to generate N weight images, each of which are then applied to a mean-shift operator [32] to locate N object position estimates from which the median values are used for the final position estimate. This framework enables the tracker to be quite resistant to partial

Figure 2.8: Centre-surround bounding box. The large red box denotes the region of interest and pixels are sampled from within. Background pixels correspond to all but the pixels within the smaller, central rectangle which denotes the foreground sample region. These samples are used for ranking features on-the-fly, as in Collins et al. [29].

occlusion, background changes caused by a moving camera and/or other dynamic objects and changes in the appearance of the object being tracked caused either by pose changes or lighting fluctuations/shadows. Although their experiments were conducted with a pool of chromatic feature types, the framework is generally applicable to other combinations of feature spaces such as LBP (Local Binary Pattern) distributions [154] or histograms of oriented gradients (HoG) [41]. The main issue with any such method is model drift - the build-up of errors over time resulting in the eventual irrelevance of the model in adequately distinguishing the object being tracked. They addressed this problem by combining appearance data from the initialisation frame in order to 'anchor' the model to a known appearance. The overall system is illustrated in Figure 2.11. While the general approach was shown to be more effective than more traditional fixed-feature approaches to tracking, the technique employed for countering model drift assumes that it is possible to obtain a good quality initialisation. A successful method for dealing with model drift will circumvent such an assumption.

Lin et al. [119] noted the work by Collins et al. [29] and generalised the Fisher Linear Discriminant method to derive a probabilistic "adaptive discriminative generative model" which they apply to the problem of object tracking under changing background and target appearance. The model is updated dynamically to deal with changing lighting and appearance. They demonstrated the method on several challenging scenarios containing pose and lighting changes. Avidan [4]

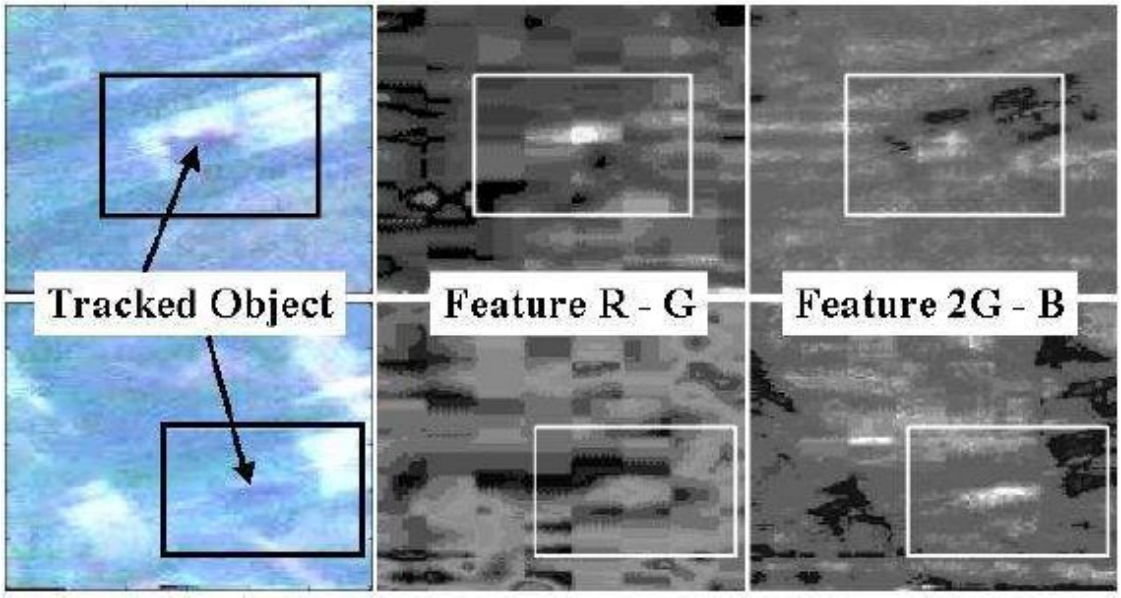


Figure 2.9: Features used for tracking an object must be adapted as the appearance of the object and background changes (Collins and Liu [30]). The source imagery (left column) is a low contrast aerial video of a car on a road. The car travels between sunny patches (top row) and shadow (bottom row). The best feature for tracking the car in sunlight (R-G) performs poorly in shadow. Similarly, the best feature for tracking through shadow (2G-B) does not perform as well in sunlight.

developed the idea by Collins et al. [29] by employing AdaBoost to learn and update and *ensemble* of weak classifiers on the fly which are then combined into a strong one as per the standard boosting approach. The ensemble is updated in each frame by training new classifiers and using them to replace the weakest classifiers from the current set. Their *ensemble tracking* algorithm is shown in Figure 2.2 and the method illustrated in Figure 2.12. To address model drift, Avidan [4] takes an effectively similar approach to Collins et al. [29] by exempting replacement of the strongest classifier generated for the first frame of tracking.

Liang et al. [115] further develop the approach by Collins et al. [29] to deal with changes in object scale, which can compound model drift issues. This is done by employing simple correlation templates to estimate object boundaries and consequently the dimensions of the bounding box to reduce overlap between foreground and background pixels in the corresponding models. They also employ the Bayes error rate as a method of ranking features and controlling the frequency of the feature selection process. Grabner and Bischof [70] and Grabner et al. [71] build upon the work by Avidan [4] by employing an on-line version of AdaBoost [61] for updating

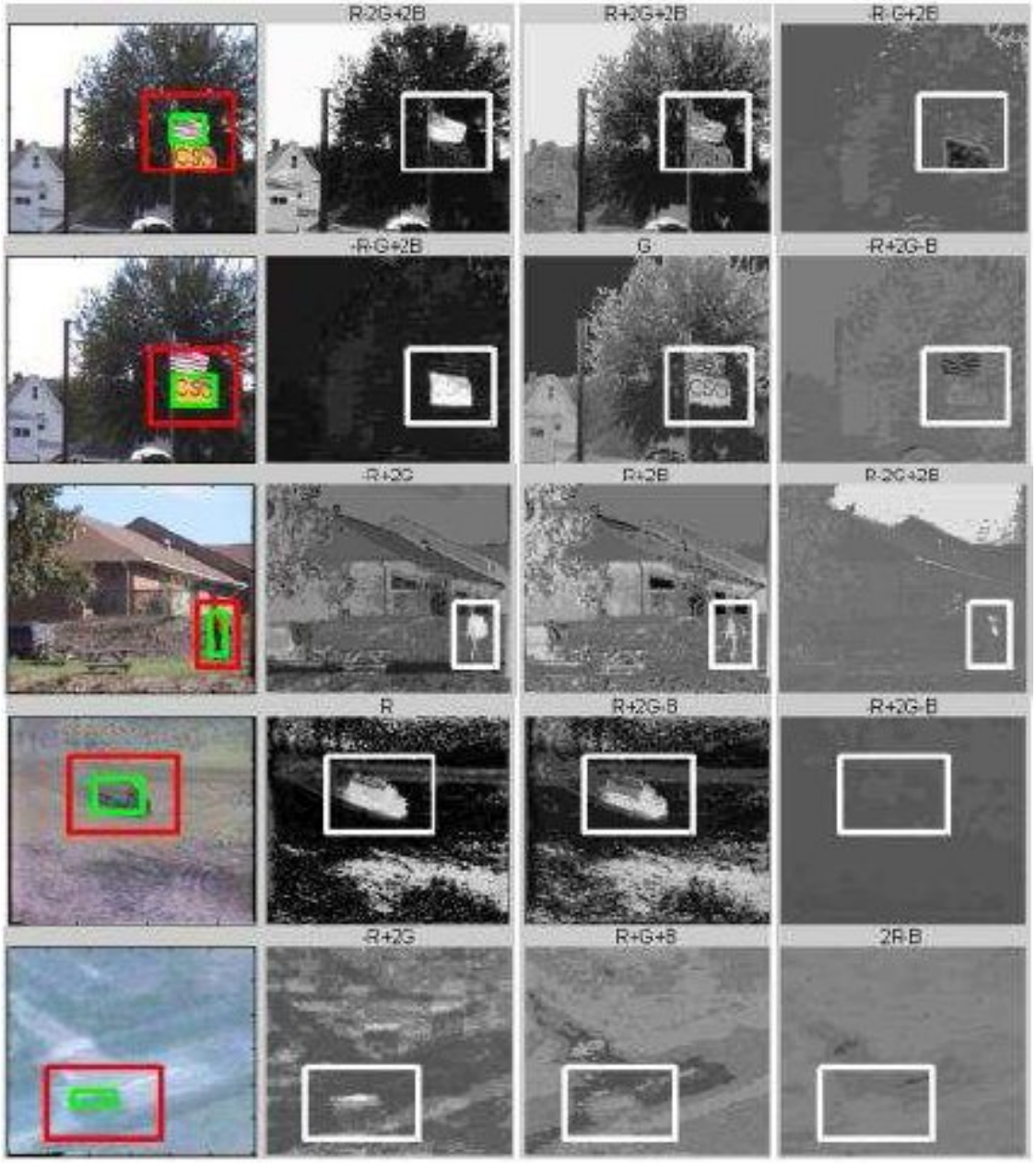


Figure 2.10: Sample video frames with ranked weight images (Collins and Liu [30]). Left column: frame with labelled object (inner box) and background pixels (outer box) pixels. Second-fourth columns: weight images corresponding to the features with highest, median and lowest variance ratio scores, respectively. The features with the higher variance ratio scores show the target object as more distinctive against the background.

an ensemble of classifiers. The algorithm they employ is illustrated in Figure 2.13. Grabner et al. [72] also develop this further to tackle the model drift problem inherent in online adaptation methods. This involves employing the SemiBoost algorithm (Mallapragada et al. [132]) as part

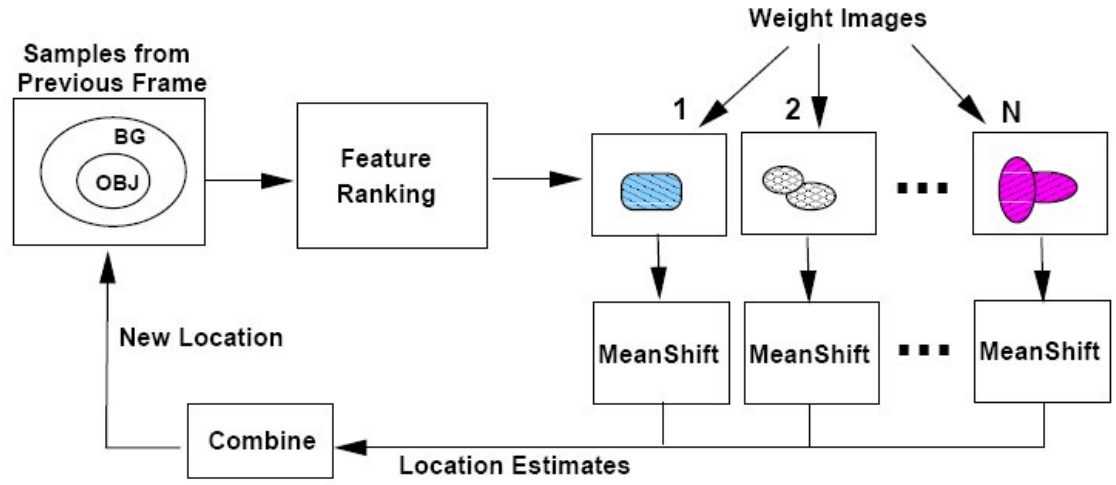


Figure 2.11: Overview of tracking system with on-line, adaptive feature selection (Collins and Liu [30]). Samples of object and background pixels in the previous frame guide evaluation of candidate features, leading to a rank ordering of features based on discriminative ability. The top N best features are applied to the current frame to compute N weight images. A mean-shift process is applied to each weight image to compute a 2D location estimate. These N estimates are combined to determine the best location of the object in the current frame and the procedure iterates.

Input: n video frames $I_1, ..., I_n$
Rectangle r_1 of object in first frame
Output: Rectangles $r_2, ..., r_n$

Initialization (for frame I_1):

- Train several weak classifiers and add them to the ensemble

For each new frame I_j do:

- Test all pixels in frame I_j using the current strong classifier and create a confidence map L_j
- Run mean shift on the confidence map L_j and report new object rectangle r_j
- Label pixels inside rectangle r_j as object and all those outside it as background
- Remove old weak classifiers
- Train new weak classifiers on frame I_j and add them to the ensemble

Algorithm 2.2: The Ensemble Tracking algorithm (Avidan [4]). An ensemble of weak classifiers is trained by AdaBoost. At each new frame, new weak classifiers are trained and used to replace the weakest ones in the current ensemble.

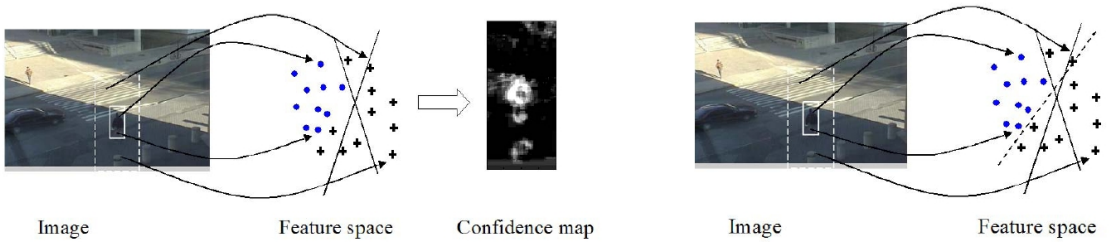


Figure 2.12: Illustration of ensemble tracking update (Avidan [4]). At each frame, the current ensemble is used to classify pixels taken from the foreground (centre) region and pixels from the background (surround) region as shown in the leftmost image. A confidence map for each pixel of the region of interest is computed (centre image) which is applied to mean-shift. The resulting fitted bounding box is used to train a new weak classifier (dashed line, rightmost image) which is integrated into the current ensemble.

of an on-line semi-supervised scheme consisting of the classification of pixels through a combination of two separate strong classifiers; one continually updated according to current image evidence and the other learned with data from the first frame acting as a static prior to discourage model drift.

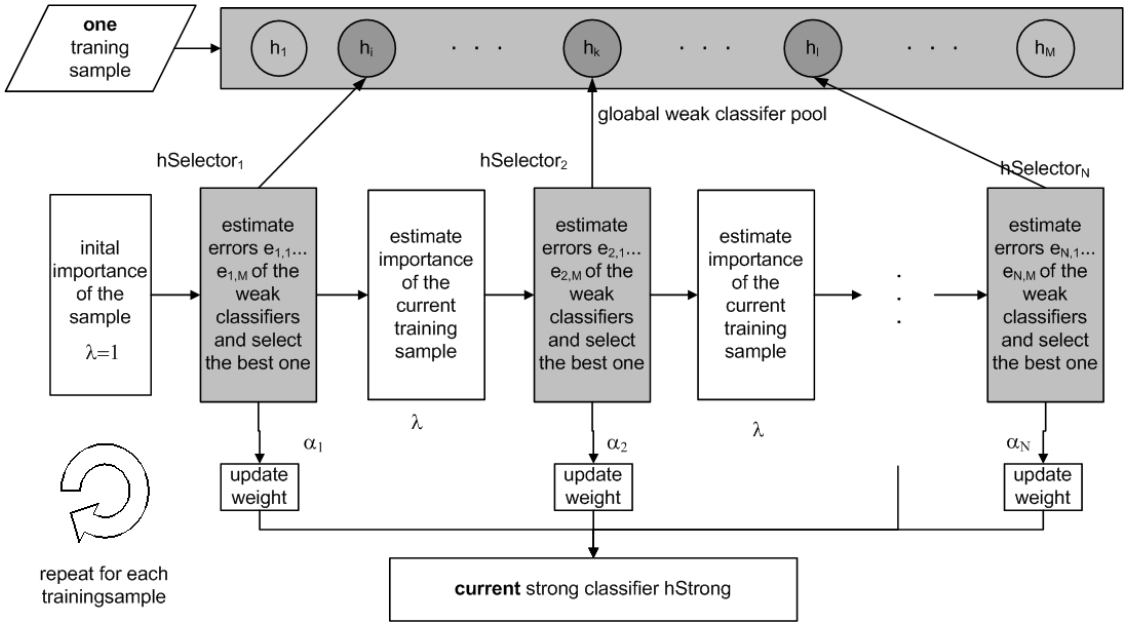


Figure 2.13: Feature selection using an on-line version of AdaBoost (Grabner et al. [71])

In other recent work, Nguyen and Smeulders [148] model foreground and background as textures represented by Gabor filter responses [90, 43, 25]. A set of discriminant functions are trained on-line using a variant of Linear Discriminant Analysis (LDA) [48] with object locali-

sation performed by maximising the sum of these functions. By using textures and maintaining a record of background textures the method was shown to be more robust to drastic changes in target appearance. Song et al. [204] perform classification as part of an on-line supervised learning scheme to disambiguate difficult situations in crowded scenes where the close proximity of objects may confuse a tracker. In this scheme, features are selected according to their strength in discriminating between individual tracked objects in order to prevent erroneous labelling caused by events such as temporary occlusions. Stolkin et al. [209] focus on integrating adaptive background models with a capability of handling cameras in motion as well as strong overlap between foreground and background colours. They describe an algorithm known as Adaptive Background Continuously-Adaptive Mean Shift (ABCshift), designed for fast and robust visual tracking under difficult conditions. Background models are continually relearned and pixel memberships computed via Bayes' rule. Their formulation ensures that colours which are shared between background and foreground are weighted lower in the estimate, with the result that only colours which are discriminative are employed in the process. Their experiments showed the technique as being capable of handling difficult conditions such as significant camera motion and fast changing backgrounds as well as variable lighting and partial occlusions. Yu et al. [244] learn changes in appearance over time (due to viewpoint and illumination changes or occlusion). This is to both track objects under such difficult conditions *and* when they reappear in the field of view after having disappeared. They take a co-training based approach which updates a hybrid, discriminative generative model online. The model employs multiple low-dimensional linear subspaces to characterise all observed variations in object appearance over time. An online support vector machine classifier is trained to focus on recent appearance variations and used for reacquiring an object following complete occlusion.

As discussed in this section, the difficulties caused by the dynamic nature of tracking lead to great fluidity in the distinguishing characteristics of a tracked object with its surrounding area. Fundamentally, it makes more sense to employ a dynamic classification approach which trades off samples of object appearance against samples of the background as well as performing appropriate sampling of both to adapt models of appearance to handle changes. In Chapter 3, we describe a system for the fast, robust tracking of multi-colour objects against changing backgrounds by employing a Bayesian classification scheme in conjunction with Gaussian mixture models of colour distributions. Using such semi-parametric models addresses some of the draw-

backs of nonparametric approaches such as histograms, namely the ability to "smooth over" gaps caused by the generally small training sets that are used as well as avoiding the need to select the number of bins. The mixture approach enables multimodal distributions to be modelled but by introducing a new model selection problem; that of choosing the number of components. We address this by describing an automatic model order selection method based on cross validation. Furthermore, we deal with the dynamic nature of colour models by encapsulating the Gaussian mixture framework in an adaptive Gaussian mixture algorithm which selectively and dynamically updates the model by sampling the tracked object and the background to deal with changes in the appearance of both target and scene caused by shadowing and lighting fluctuations.

The feature-selection approaches described above are a natural extension to the above idea, and enable multiple feature types to be considered as part of the general tracking problem. This constitutes a natural extension of spatial sampling for tracking to the "featural" domain from the Spatial-Featural-Volume (SFV) (see Figure 1.8). Previous work has generally not addressed the problems of dynamic feature selection for tracking small, difficult targets against highly cluttered and dynamic backgrounds, such as pedestrians in a wide-angle view of a crowded scene, which can impact on classification performance and compound model drift problems. Further, the model drift problem is generally addressed by maintaining and combining a static model intended to reflect a "reference" representation with the dynamic one. Such static models are inherently assumed to reflect a reliable appearance of the target and can quickly become irrelevant under long-term changes. In Chapter 4 we describe a framework which improves upon the previous methods by dynamically modelling the reference model for each feature domain in order to facilitate a more complete adaptive approach without prior assumptions. By ensuring each domain is kept disjoint, a more appropriate selective adaptive methodology is provided. Further, we employ a feature ranking methodology designed to reduce redundancy amongst features (see Section 2.1) used in order to improve classification performance. This constitutes an approach for filtering slices from the corresponding SFV.

2.3 Sampling for recognition

Visual recognition encompasses several type of association tasks such as recognising a class of objects from an instance in an image, determining a gesture being performed by an individual in a video stream [161, 89] or recognising a specific person from visual characteristics such as

a static face image [226, 251] or their gait in motion [65]. The detection of certain entities also fall under this category. As with tracking, the notion of object association underpins the process, which involves the sampling of image features and their comparison with expectations dictated by a model. The kinds of features employed are typically hugely varied and can consist of both combinations of raw image properties such as brightness or RGB values as well as transformations of various kinds. These include Principal Components Analysis (PCA) transformations of vectors of raw pixel brightnesses (e.g. for face recognition [226]), Gabor wavelet transforms that model the response profiles of simple cells in the primary visual cortex of the brain (e.g. for 3D object recognition [237]), colour histograms (e.g. [212]) which encapsulate the photometric distribution of object pixels and may be used for indexing, histograms of oriented gradients (HoG) [41] which form the distribution of edges and their orientations for a class of objects and which have been used for pedestrian detection and the scale invariant feature transform (SIFT) [121] which is used for various recognition tasks and consists of a four-stage filtering process to recover robust features which are invariant to various transformations of the object such as pose and scale. Typically, the most useful sets of such features for discrimination encapsulate a specific spatial structure reflected by the real-world geometric characteristics of the object(s) concerned. All of these methods involve sampling the image or transformations of the image at pixel or subpixel level. We then consider the sampling process for recognition as inclusive of a filtering component which rejects features that do not embody the discriminative requirements of the task.

In recent years, Local Binary Patterns (LBP) features have been extremely popular as a computationally inexpensive yet effective type of feature for statistically sampling and modelling local fluctuations in images both spatially and temporally. They were initially developed as an extension to local contrast measures for texture discrimination (Ojala et al. [154]). Since then, LBPs have been used either as a primary or supporting technique for a huge variety of applications such as object detection [246], object classification [3], automated industrial inspection [131, 120, 151], underwater image classification [205, 26, 14], aerial image segmentation [227], guiding active contours for texture segmentation [187], medical tissue analysis [171], multispectral image segmentation [125], colour-constant image indexing [34], mobile robot navigation [44, 144], steganalysis [106], design of information displays [56], overhead person recognition [27], face recognition [1, 2, 247, 250, 113, 183], face detection [92], facial expression recognition

[53, 194, 193, 116], gender classification [210, 114], head pose estimation [126], iris recognition [211], palmprint recognition [229], text detection [239], person detection and tracking [202], activity analysis and recognition [98, 99], eye gaze tracking [122] and multiple object tracking [214].

2.3.1 Local Binary Patterns

A pattern is extracted from a local neighbourhood surrounding each pixel ξ in an image and is defined as a series of K Boolean values (or *textons* t_k) indicating the sign of the K surround intensities $x_{k,\xi}, k = 0..K-1$ thresholded by the centre pixel value y_ξ. The Boolean values are then treated as a binary number and translated into a single decimal value w_ξ for the pattern at that pixel (see Figure 2.14):

$$w_\xi = \sum_{k=0}^{K-1} t_k^k, \;\; t_k = \left\{ \begin{array}{l} 1, \;\; (x_{k,\xi} - y_\xi) > 0 \\ 0, \;\; otherwise \end{array} \right\} \tag{2.17}$$

For each image, histograms of these decimal pattern values w_ξ are formed. Consequently, for these patterns derived from eight pixels in the surround, the histograms cater for 2^8 possible decimal values, resulting in 256 bins. Individual image histograms are averaged over all the images of a class. The underlying assumption is that such averaged pattern distributions will be sufficiently different between classes of textures for the reliable matching of a single image on the basis of its LBP histogram.

This form of local binary pattern is largely robust against global illumination changes (since surround pixels should remain either higher or lower than the centre value except in extreme cases) but highly sensitive to rotation. Ojala et al. [155] extended the method to arbitrary *circular* neighbourhoods surrounding each pixel with parameters R defining the radius and P the number of points equally-spaced around the circumference rather than at specific pixel positions (see Figure 2.15). For this approach, intensities at sub-pixel positions are computed using interpolation. Experiments were done with $P = 8$ sampled points around the centre at a fixed distance of $R = 1$ pixel widths. This extension permits a more robust multiresolution approach to texture analysis as well as providing amenability to rotational invariance.

Rotational invariance with such a scheme is achieved by simply "rotating" an extracted pattern to its smallest possible value (Pietikäinen [165]). For multiresolution analysis, several circular neighbourhoods may be considered at different values for R and P, each of which results in

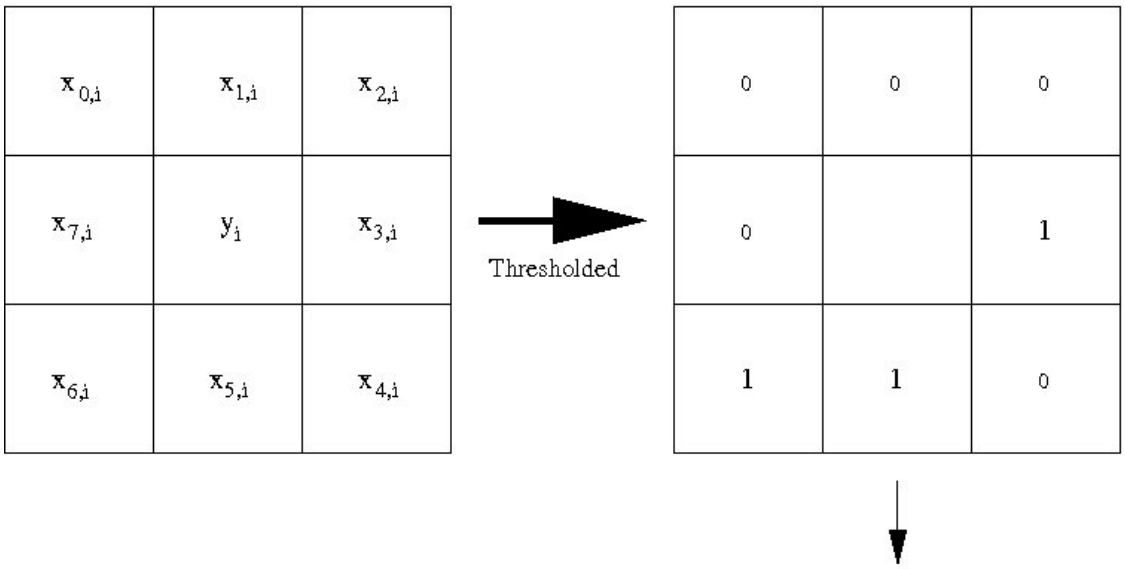


Figure 2.14: Local binary pattern (Ojala et al. [154]). For each pixel, intensity values in the neighbourhood are thresholded by the centre value and transformed into a binary string according to the resulting sign. The decimal equivalent of the string is used as the feature value for that pixel.

a separate histogram of equivalent decimal values. In order for features to remain representative as well as reasonably dense as a description of local texture, practicalities of limited resources dictated a limit of three such circular neighbourhoods (or *predicates*) at radii of $R = 1, 2$ and 3 with corresponding sample sets at $P = 8, 16$ and 24 points. Rather than attempt to model these distributions jointly which would result in a prohibitively huge histogram of 2^{48} bins, they are modelled individually with the three histograms subsequently concatenated for a final descriptor ((Mäenpää et.al [130], Mäenpää and Pietikäinen [129]). Although this has the disadvantage of statistically decoupling the scales, it has the benefit of alleviating the problem of the curse-of-dimensionality and the corresponding requirement for impractically large data sets.

To further reduce the size of histograms, an experimental example of heuristic feature selection was performed by Ojala [155] who related observed featural characteristics to physical realities by showing that the majority of extracted binary patterns in images corresponding to certain types of feature such as spots, edges and corners exhibited a limited number of *transitions* from one to zero and vice-versa in a binary string. Around ninety percent of these so-called *uniform* patterns had no more than two such transitions with the result that an eight-bit LBP pattern could take a maximum of 58 possible string configurations. Consequently, the 256-bin

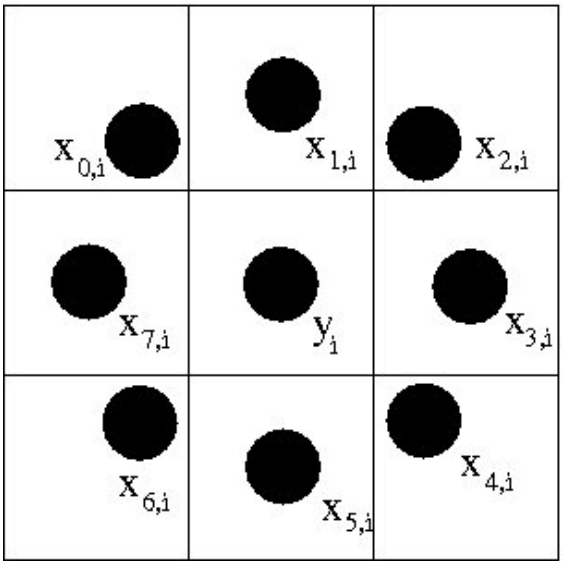


Figure 2.15: Circular local binary pattern (Ojala et al. [155]). For each pixel, intensity values are sampled at subpixel level at a fixed distance and regular angular points surrounding the centre. Similar to the approach from Figure 2.14, they are then thresholded by the centre value and transformed into a binary string according to the sign of the result. The decimal equivalent of the string is used as the feature value for that pixel.

histogram could be trimmed to 59 bins (with the 59th containing all non-uniform patterns). The effect on classification was shown to be minimal or even beneficial as a result. The resulting circular, rotation-invariant uniform operator (or class of operators) is generally identified by the label $LBP^{riu2}_{P,R}$ with P denoting the number of sampled points, R the radius in pixel widths, *ri* indicating rotation invariance and $u2$ denoting the use of uniform patterns. Multipredicate operators are specified (for example) as $LBP^{riu2}_{8,1+16,2+24,3}$.

To improve the spatially descriptive extent of LBPs for multiresolution analysis Mäenpää and Pietikäinen [127] used low-pass Gaussian filtering so that single point samples would incorporate information integrated over a larger area. Colour-opponent versions of the LBP, known as opponent-colour local binary patterns (OCLBP) were also developed (Mäenpää and Pietikäinen [128]) which involved the extraction and thresholding of point samples from different colour channels rather than single intensity images.

2.3.2 Further developments of LBP

In its basic form, the LBP method can result in histograms with large numbers of bins which in turn result in restrictions in the spatial support area from which information can be collected as well as the requirement for large amounts of data for a representative model. As such, methods for reducing the complexity of LBP models are necessary for alleviating these limitations. The use of uniform patterns was a heuristic step designed to reduce the set of patterns to those most useful and consequently also reduce the complexity of resulting models while minimising any loss of descriptive power. Other drawbacks include sensitivity to local noise such as pixel quantisation errors and local nonmonotonic illumination fluctuations.

To address some of the natural limitations of the standard circular multipredicate approach such as the disjoint nature of the distributions for individual predicates, Mäenpää and Pietikäinen [127] made use of cellular automata for encoding patterns over a larger spatial area of the neighbourhood. The Multi-scale Block LBP (MB-LBP) (Liao et al. [118]) was designed to overcome the short spatial support area of traditional LBP which results in sensitivity to noise and prevents the incorporation of larger-scale structures. This involved effectively subsampling the image by averaging over regions (blocks) and treating each result as a single value for LBP computation (see Figure 2.16). In doing so, larger areas may be taken into account and results are less sensitive to local noise. They also used AdaBoost to select the most reliable uniform patterns from a collection of MB-LBPs at different scales and applied the method to face recognition. The method was also used for face detection by Zhang et al. [248].

Lahdenoja et al. [107], motivated by greater computational efficiency in LBP-based face detection and recognition, looked at reducing feature vector lengths by examining at the property of "symmetry" in LBPs, defined as the minimum of the number of ones and the number of zeros. In the context of face recognition, high-symmetry patterns were experimentally shown to correlate strongly with the most discriminative parts of facial features such as the eyes. They proposed generating LBP histograms on the basis of symmetry levels for different patterns and showed greater relevance for facial discrimination than simple uniform patterns alone on the FERET face database. A feature-selection approach was taken by Liao et al. [117] who simply trimmed the most infrequent patterns from a histogram of all possible patterns. Shan et al. [193] proposed boosting LBP classifiers for facial expression recognition. They developed Conditional Mutual Information based Boosting (CMIB), derived from the Conditional Mutual Information Maximi-

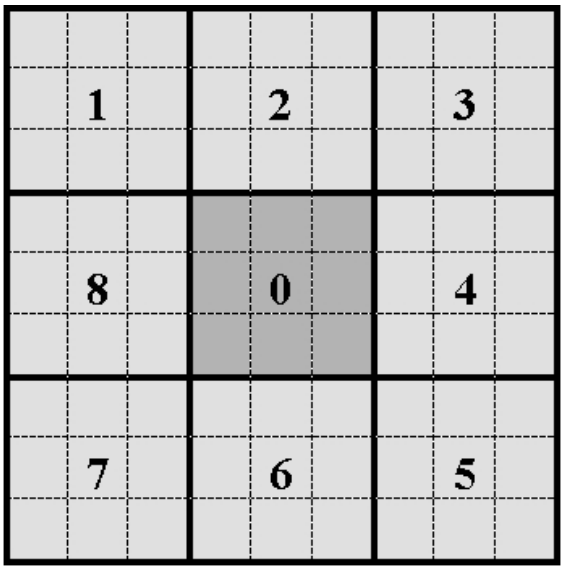


Figure 2.16: The Multi-scale Block Local Binary Pattern (MB-LBP) (Liao et al. [118]). Blocks of pixels are averaged over before being treated as a single value for LBP computation (here, each value comprises the average of a local 3x3 neighbourhood). This amounts to a subsampling of the image and enables more robustness against local noise as well as effectively increasing the spatial support area of derived patterns.

sation (CMIM) algorithm for selecting mutually complementary features with low redundancy [55], to learn a sequence of the most relevant weak LBP classifiers derived from a collection of sub-regions at different scales. Those selected were combined into a strong single classifier. They also showed the benefit of using CMIB over AdaBoost for the same task (see Figure 2.17). Shan et al. [196] employed AdaBoost to select specific bins from histograms and use them as weak classifiers rather than take them as a whole. These were combined into strong classifiers which comprised the most discriminative bins. The effectiveness of the resulting LBP-Histogram (LBPH) method was demonstrated in a facial expression recognition task on an established face expression database.

To improve the relationship between LBP patterns and physical image structures such as corners, Jin et al. [92] developed the Improved LBP (ILBP) by effectively doubling the number of patterns for a fixed support area. This was achieved by employing the mean value of the region as a threshold rather than the central pixel value and including the thresholding central value as part of the pattern. As a result, features such as corners could be more readily reflected in extracted patterns. They employed this technique for face detection by modelling face and

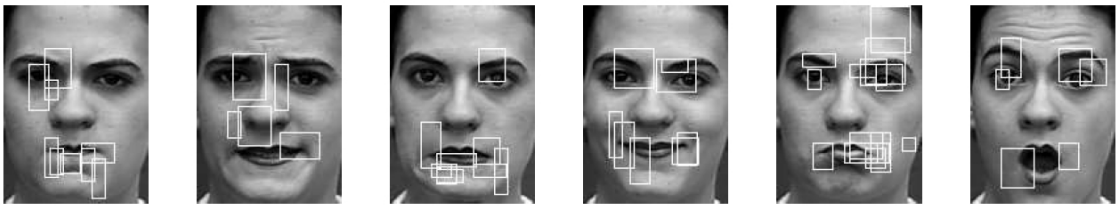


Figure 2.17: LBP patterns selected by a boosting procedure for facial expression recognition (Shan et al. [193], Shan and Gritti [196]). Each rectangle denotes a facial region corresponding to a selected weak LBP classifier trained on that region. They can be viewed as corresponding to key physical locations useful for discriminating between expressions.

non-face classes with multivariate Gaussians and using a Bayesian classifier for discrimination.

Zhang et al. [249] developed the Local Gabor Binary Pattern Histogram Sequence (LGBPHS) representation for representing faces. This method was intended to avoid the problems of generalisation that accompany statistical learning methods by incorporating multi-scale and multi-orientation Gabor filters for the decomposition of normalised face images prior to the application of LBP operators. These Gabor Magnitude Pictures (GMPs) are then processed by LBP operators to create Local Gabor Binary Pattern (LGBP) maps. The maps are segmented into multiple non-overlapping regions and histograms generated for each segment. Finally, all histograms are sequentially concatenated to form an LGBPHS representation for the individual (see Figure 2.18). Experiments showed an impressive robustness against significant fluctuations caused by illumination, expressions and time gaps between images as well as requiring a single sample to generate the representation. Xie et al. [238] built on this method by proposing a scheme known as Volume-based Local Gabor Binary Patterns (V-LGBP) for representing and recognising faces. The Gabor filtered images are here combined into a volume where the third coordinate determines the particular Gabor filter used. The relationships between individual filter outputs are then encoded by LBP to characterise local saliency.

Other recent developments of LBP methodology include Local Ternary Patterns (LTP) (Tan and Triggs [215]), consisting of patterns comprising three values $\{-1,0,1\}$. This generalised LBP to deal with severe local illumination changes such as shadowing and to reduce sensitivity to image noise by introducing a threshold ε for the centre value x of a predicate. A surround pixel value y within this threshold, $|y-x| < \varepsilon$ either above or below the centre, translates to 0 in the resulting pattern with values above indicated by a 1 $y \geq x+\varepsilon$ denoting a 1 and values below

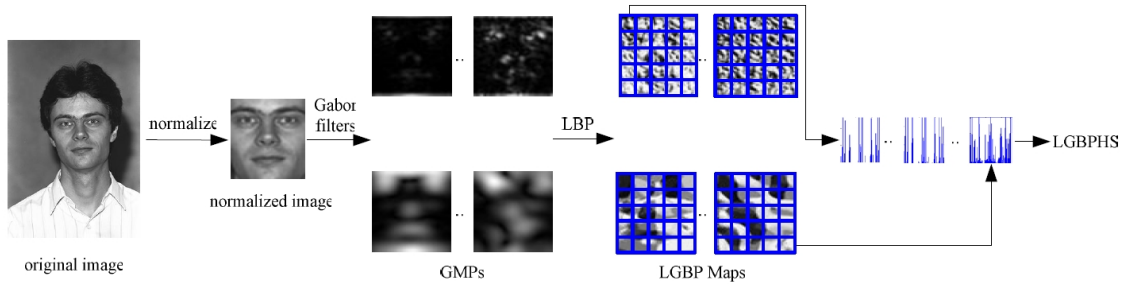


Figure 2.18: The procedure for generating a Local Gabor Binary Pattern Histogram Sequence (LGBPHS) (Zhang et al. [249]). Normalised face images are Gabor filtered at multiple scales and orientations to derive Gabor Magnitude Pictures (GMPs) which are then processed with LBP operators to generate Local Gabor Binary Pattern (LGBP) maps. These maps are dissected and histograms of patterns formed for each segment. The histograms for all LGBP maps are sequentially concatenated for the final LGBPHS representation.

$y \leq x - \varepsilon$ becoming a -1. These were tested on established face-recognition datasets containing significant illumination variations.

Zhou et al. [254] noted that the standard approach of retaining uniform patterns is sensitive to noise and results in the discarding of useful information. They derived an extended LBP operator from the analysis of nonuniform pattern structure and statistics and experimentally demonstrated greater robustness against noise for a texture discrimination task. He et al. [83] developed a Bayesian LBP (BLBP) operator as a texture descriptor as part of a Filtering, Labeling and Statistic (FLS) framework which was used to derive pattern labels as a probabilistic optimisation procedure. These patterns helped to reduce sensitivity to noise.

2.3.3 LBP-based recognition

LBP methods have been used for a multitude of recognition tasks in addition to the original texture classification motivation [154]. For example, Cohen et al. [27] employed them to characterise textures as part of an overhead-view person recognition system. Sun et al. [211] employed them to model the texture of iris images for iris recognition. They combined them with graph-matching techniques for classifying the structural characteristics of the iris images. Wang [229] modelled the textures of palmprints for recognition by deriving LBP histograms from subwindows of palm images and employing AdaBoost to select the most discriminative amongst them to good effect. In recent times, however, facial identity and expression recognition have

perhaps been the most popular areas for the application of LBP methods.

Face recognition was addressed by Ahonen et al. [1, 2] who modelled facial identity by dividing the face into small regions and modelling LBP statistics for each one (see Figure 2.19). The histograms for each region were then averaged over all examples for each individual, concatenated and used as a single descriptor of identity. Tan and Triggs [216] combined Gabor wavelets with LBP for face recognition. Whilst LBP is suitable for modelling fine details, Gabor wavelets can encode characteristics such as face shape at coarser scales, making the two complementary for fusion at the feature level. The technique involves extracting Gabor and LBP features independently and performing PCA on the results for dimensionality and noise reduction. The feature vectors are fused by concatenation and normalised before applying a kernelised version of Linear Discriminant Analysis (LDA) called Kernel Discriminative Common Vectors (KDCV) to extract the optimally discriminative nonlinear features from the fused vectors. (see Figure 2.20). The benefit of combining these two features was demonstrated on challenging face databases.

Figure 2.19: Face recognition using LBP (Ahonen et al. [2]). The face image (left) is preprocessed (middle) and segmented into multiple regions (right). LBP histograms are generated for each region and averaged over all samples for the individual. The resulting average histograms are concatenated to derive a descriptor for the individual.

Feng et al. [53] employed a similar approach to [1] for the recognition of facial expressions. Face images were divided into multiple non-overlapping regions with LBP histograms generated for each one. These were concatenated into a single descriptor of the face, averaged over all examples of the corresponding class. They used a linear programming technique to find separating hyperplanes to perform classification between each pair of seven identified expressions. A tournament tree was employed for final decision making. Shan et al. [194] employed weighted

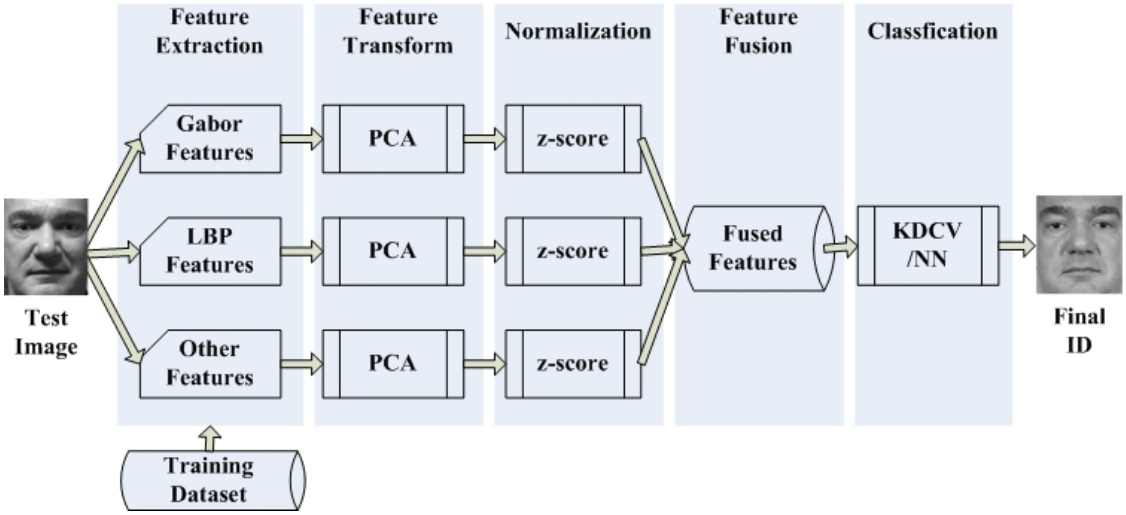


Figure 2.20: The combination of Gabor wavelets with LBP for face recognition (Tan and Triggs [216]). The method involved the use of dimensionality reduction via PCA on feature vectors for each method individually, followed by concatenation for fusion and the application of Kernel Discriminative Common Vectors (KCDV) for the extraction of optimally discriminant nonlinear features.

Chi-square statistics and Support Vector Machine for matching the LBP histograms for face regions. The same authors [193] proposed a novel learning algorithm for boosting LBP patterns to further improve classification performance. Liao et al. [116] also addressed the facial expression recognition task by segmenting face images into weighted regions denoting key facial features such as the eyes, lips and nose. They then derived LBP histograms from both intensity and gradient maps to capture high-frequency textures and the Tsallis entropy of Gabor filter responses for low to mid-frequency structures for each region. A null-space based Linear Discriminant Analysis (NLDA) step was used for determining discriminative global expression features. All three features were then combined for classifying expressions.

An area where LBP methodology can be improved is in more rigorously defining the process by which samples are taken from the support region for a reference point (e.g. the centre pixel). In the original methodology, as well as most of the more recent developments that are built upon it, samples are taken with a fixed spatial topology such as pixels in a rectangular surround region or subpixel samples at fixed radii from the reference. Within the context of the Spatial-Featural Volume, each individual component of a binary pattern may be considered an individual feature. For example, for a 3x3 mask eight LBP binary images may be derived with each one representing

one of the thresholded pixels from the surround region for each pixel (see Figure 2.21). Previous methods derive patterns by linearly encoding *all* of the N binary images for N-bit LBPs, with the result that histogram bins are derived from joint statistics of fixed circular topologies that themselves may contain counterproductive or redundant elements. Moreover, such topologies are applied in an unchanging manner in all classification situations whereas different topologies will be more relevant depending on the classes of object being compared. More recent work performing selection of patterns or bins for improved discrimination operate after the complete encoding takes place.

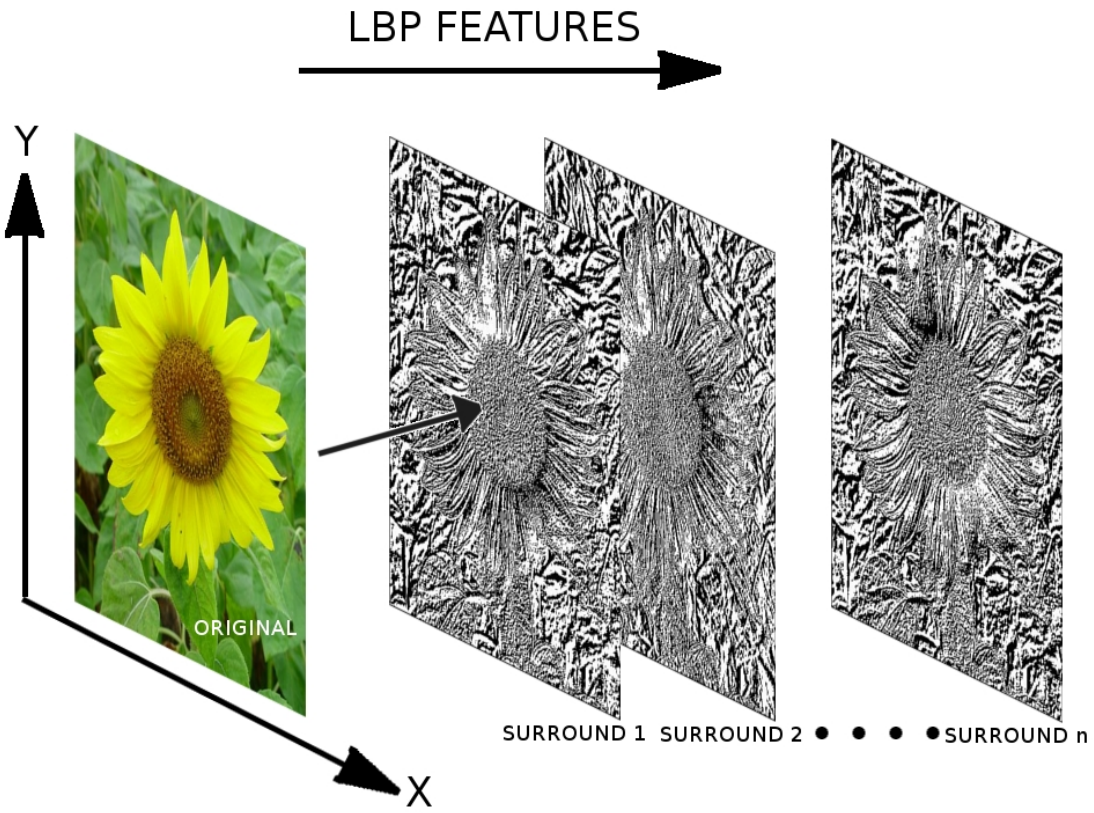


Figure 2.21: Spatial-Featural Volume for the LBP features of an image. Each slice constitutes the binary-thresholded pixel from a specific position from the surround (e.g. 8 slices for an $N = 8$ LBP derived from 3x3 pixel regions surrounding each pixel). Previous methods encode all slices simultaneously without considering those which are most discriminative for classification.

A more appropriate, rigorous and robust approach to formulating sampling strategy for LBP methods is to *derive* the most relevant and effective local topology for the extraction of these patterns in a manner which is flexible and contextually class-dependent. In other words, LBP sampling strategy should be determined through a feature selection approach which selects the

most relevant "slices" from the LBP SFV for an image. This constitutes a sampling scheme with an adaptive, dynamic strategy which naturally solves the problems of limited spatial support (since samples may then be gathered from arbitrarily large neighbourhoods) and enables fully-coupled statistics without prohibitively large training sets across multiple scales as well as improving discriminative power. As such, this is arguably a more rigorous way of improving the discriminability of LBP-based classifiers. We derive a general framework for achieving this in Chapter 5 which is largely computationally inexpensive and naturally applicable not only to the traditional LBP framework but to many of its extensions and derivatives without modification.

2.4 Summary

In this chapter, we have examined the issues of feature selection for classification, visual tracking and visual recognition within the context of visual sampling. As discussed in Chapter 1, the concept of evaluating pixels and ranking features are directly analogous, and conceptually unified when viewed within the concept of the Spatial-Featural-Volume of an image (Figure 1.8), where the first two dimensions correspond to the spatial coordinates of an image and the third dimension corresponds to the "featural" domain (for video, a fourth dimension may be assigned as temporal). Both involve assigning some "goodness" value, either to pixels in the spatial domain or to features. Pixels may be considered in isolation (e.g. Stauffer and Grimson [206] for modelling backgrounds on an individual per-pixel basis) or in spatial conjunction with one another (e.g. Russell and Gong [185] for estimating background pixels while taking into account their surrounds). Similarly, features may be ranked in isolation (e.g. using AdaBoost [61]) or in conjunction with one another (e.g. the CMIM algorithm [55]). Given such measures of "goodness", pixels *or* features may be taken as weighted combinations or distinct subsets chosen. As such, the notion of feature selection fits naturally into the concept of visual discrimination.

In Section 2.1 we explored some theoretical concepts in feature selection and their practical realisation in the form of various algorithms. Selecting small discriminative subsets of features with minimal descriptive redundancy improves performance and efficiency in terms of computation time and storage. In general however, it is computationally prohibitive to find the *optimally* most discriminative set of features given some data, known as the Markov blanket (Pearl [162]). Consequently, most algorithms attempt to find computationally cheaper suboptimal solutions and generally fall into the categories of: *wrapper*, algorithms that perform cross validation using a

specific classification method as part of the process of feature selection; and *filter*, classifier agnostic methods that are generally faster but not necessarily optimal for all classification methods. We compared these two classes of feature selection algorithm and showed how boosting methods such as AdaBoost [61] have been frequently used as a wrapper method for feature selection and more recent filter techniques such as CMIM [55] to find more robust and less redundant subsets of features using a sound information-theoretical criterion as an approximation for estimating redundancy.

In Section 2.2, we discussed previous tracking methodologies by addressing three main areas: (a) tracking by modelling foreground only; (b) modelling and removing the background in order to determine foreground regions for tracking; and (c) performing classification between foreground and background for tracking. Given the inherently fluid relationship between foreground and background, we argued that classification based approaches were the most natural way of addressing the problem. Furthermore, the very fluidity that necessitates this approach also suggests the need for the ability to adapt models to deal with changing appearance. A further extension of the argument incorporated the benefit of selecting *features* on the fly depending on the most useful method of discrimination at any given time during a tracking task such as colour or shape. We discussed some recent, powerful techniques for achieving this along with two common major limitations of such techniques: (a) the ranking of features using relatively limited methods such as variance ratio or AdaBoost based selection; and (b) the model drift problem and the use of static reference samples as a rudimentary method to alleviating it. Accordingly, in Chapter 3 we describe a framework for modelling complex multi-colour distributions of foreground and background using Gaussian mixture models of colour for real-time tracking. These models overcome some of the problems with non-parametric histogram-based distribution representations by smoothing over gaps caused by a sparsity of training data as well as preventing issues of choosing bin sizes. Multimodal distributions may be modelled using such a scheme but introduces a model order selection problem; namely, the number of Gaussian components required. We describe a method for automatically selecting model order. Furthermore, we integrate an adaptive method to enable the mixture models to adapt on-line from samples dynamically gathered on-the-fly from the image. This enables tracking to perform robustly under non-uniform illumination changes. In Chapter 4 we extend the work in Chapter 3 to incorporate the featural dimension for object representation. Accordingly, we describe a framework for (a) improving the ranking approach

to reduce redundancy amongst selected features on-the-fly for difficult tracking problems and (b) enabling dynamic reference samples through selective updates individually for each feature to obviate inappropriate assumptions by previous methods and further alleviate the model drift issue. We show the benefits of these contributions by applying the method for tracking small pedestrian targets in highly cluttered scenes in extremely difficult situations.

In Section 2.3, we covered in some depth previous Local Binary Pattern related methods for recognition tasks. Having described the basic LBP methodology, we focused on some key extensions with special consideration given to methods designed to improve classification performance. Previous methods for enhancing the discriminative capacity of LBP models include selection processes for choosing the most useful patterns. However, in all these cases, the patterns are encoded from *all* the individual LBP samples (slices in the Spatial-Featural-Volume (SFV), see Figure 2.21). Having considered the problem in terms of the SFV, a more natural way to select patterns is to perform feature selection *prior* to pattern encoding (i.e. choose the slices from the SFV and encode the results). This approach has several benefits and solves several problems in a natural way: (a) Eliminates the restrictions on spatial support for traditional LBP approaches, (b) permits the modelling of joint statistics over larger areas without requiring prohibitive quantities of training data, (c) chooses areas of focus depending on the two classes being compared, hence constituting a more natural, flexible and dynamic sampling strategy and (d) can be used with the traditional method and many of its more recent extensions as an "add-on". Accordingly, in Chapter 5 we develop a framework for LBP-based visual association that exhibits these properties. In doing so, we develop a novel, computationally efficient filter specifically for binary feature selection called Binary Histogram Intersection Minimisation (BHIM) which is experimentally shown in comparison with both AdaBoost and CMIM on both synthetic and real data to (a) employ varying computational resources depending on the training set, with good quality features being found far more quickly than low quality ones and (b) find stronger subsets of low-redundancy binary features than either CMIM or AdaBoost. We build sparse models from these chosen features and experimentally show their improvement in classification performance over standard LBP methods in face and texture recognition tasks. We call these models Multi-Scale Local Binary Features (MSLBF).

Chapter 3

Colour Feature Sampling for Tracking

The visual tracking of an object through a scene is rooted in the notion of object association. From frame to frame of an image sequence, the task is to determine the state or states of the object being tracked, e.g. 3D pose, orientation, scale or position in the image. Scale and position are often estimated from the region of the image containing the object, with these states approximated by fitting bounding boxes. The pixels within the bounding boxes are generally identified as associating strongly with the object given some internal model of expected object appearance. The goal is to collect the most relevant pixels as samples for estimating these object states. In terms of the generalised notion of adaptive visual sampling, the tracking process can then be viewed as generally comprising two complementary filtering components forming part of an overall *sampling strategy*:

1. A predictive mechanism acts as a method for narrowing the search for the object. This may be established to varying degrees of sophistication. In the simplest case, a priori heuristics may be employed; in any given frame t of a tracking sequence, samples may be collected within a fixed range of the position of the object in frame $t-1$ with the range fixed according to various factors, such as the frame-rate and maximum expected velocity of object movement. Ultimately this may be naturally extended to a more sophisticated, empirical, dynamic sampling strategy which incorporates several factors such as direction of object movement, camera movement, frame rate, relative object velocity and changes in object size. Data association techniques include components for predicting such states

(e.g. Kalman filters [94, 230] and particle filters [69, 166]) which in turn may be used to narrow the search for evidence and reduce effort at the next stage.

2. The samples (pixels) collected are filtered according to their associational strength with the object of interest. The process of doing this involves the computation of a matching measure between each sample and the model with samples subsequently accepted or rejected accordingly for state estimation. For example, samples may be evaluated according to the probability that they were generated by the object. In this case, "acceptance" is a varying quantity denoted by the probability value with "rejection" indicated by zero probability. As such, the model may be viewed as the basis for the filtering process. The aim is to obtain a set of samples which are considered representative of the object. They may then be used to compute more accurate empirical estimates of object state. Appropriate data association techniques combine these empirical estimates with predictions to derive more optimal final estimates.

In this chapter we develop a methodology for the use of colour features as a fast cue for real-time tracking applications. Colour features have been used for a variety of tasks such as segmentation [201], tracking [138] and object recognition [134, 212]. We describe previous colour-based methods appropriate to this area and address specific limitations which are; (a) The use of non-parametric models which can be sensitive to small training sets and (b) A lack of a dynamic sampling strategy which prevents object models from maintaining relevance under changing conditions over time. In doing so we develop a sampling strategy framework for tracking based on pixel feature statistics and apply it for colour. This involves; (a) The use of semi-parametric Gaussian mixture models to capture multi-colour distributions, (b) An algorithm for automatically selecting the number of components of a Gaussian mixture model, (c) A fast method for real-time tracking of multi-colour objects, (d) A Bayesian formulation for context-dependent pixel classification and (e) A mechanism to facilitate a dynamic sampling strategy by adapting models on-line.

3.1 Scope of the problem

A prime concern for any object association task being conducted over time is the continuing relevance of the internal model. In most real-world situations, dynamic conditions such as viewpoint changes and lighting fluctuations cause transience in the strength of association between models

and objects of interest. This problem is particularly acute for tracking tasks in uncontrolled environments, where static models may lose their relevance frequently and rapidly. Consequently, internal models need to be adapt accordingly; in other words, the sampling strategy for tracking should be flexible and dynamic.

3.1.1 Modelling colour distributions

Colour can be a very effective, strongly salient and computationally cheap visual cue for object association. Swain and Ballard [212] described a scheme which used histograms for modelling the colours of an object. The colour space was quantised through the histogram's structure which comprised a number of "bins". An algorithm known as "histogram intersection" was used for matching image histograms with model histograms. Although colour histograms can be used to approximate distributions in colour space, the level of quantisation imposed on the colour space influences the resulting distribution. If the number of bins n is too high, the estimated distribution will be "noisy" and many bins will be empty. If n is too low, distribution structure is smoothed away. Histograms are effective only when n can be kept relatively low and where sufficient data are available. A potentially more effective semi-parametric [12] approach for colour distribution estimation may be found through the use of Gaussian mixture models. With this approach, a number of Gaussian functions may be taken as an approximation to a multi-modal distribution in colour space. Finite mixture models been previously discussed at length [12, 52, 142, 168, 169, 170, 221] although most of this work has concentrated on the general studies of the properties of mixture models rather than developing vision models for use with real data from dynamic scenes.

In Section 3.2 we describe the use of Gaussian mixture models for modelling the colour distributions of multi-coloured objects. They are used as part of a probabilistic approach to our tracking framework's filtering-based sampling strategy, involving the computation of conditional probabilities to evaluate image samples' association with the model.

3.1.2 The model order selection problem

The use of colour mixture models in dynamic scenes is not without its difficulties. A common problem associated with density-based modelling of statistical data involves the selection of the number of parameters for a model, known as the *model order selection* problem [12]. With colour mixture models, this involves the selection of the number of Gaussian components. The goal is

to generate a model that provides accurate predictions for new data. Too few parameters can lead to a poor model which over-generalises the data (high bias), while too many parameters can result in an overfit of the model to the training data (high variance) [67]. In either case, the underlying distribution responsible for the training data is not reflected accurately and performance on new data will be poor. Existing methods for model selection are usually rather *ad hoc*. An exception is the recursive algorithm of Priebe and Marchette [169]. It was extended to model non-stationary data series through the use of temporal windowing. Their algorithm adds new components dynamically when the mixture model fails to account well for a new data point.

Here, to simplify the problem and corresponding solution we separate the issues of selecting model order and adapting them on-line. In Section 3.3 we describe an iterative algorithm for automatically determining model order for a Gaussian mixture based on a fixed data set. Components are added incrementally whilst cross-validation is used to monitor generalisation ability and prevent overfitting.

3.1.3 Tracking with colour cues

Methods have been proposed for colour-based detection and tracking of skin-coloured objects (e.g. [100, 188, 189, 235]). In particular, a system constructed by Wren *et al.* [234] enabled tracking of entire people in controlled environments with static cameras. Each pixel in an image had an associated feature vector comprising spatial and colour components. These feature vectors were clustered, which led to a collection of "blobs" defined by spatial and spectral similarity. A collection of blobs constituted a representation of a person. This limited tracking to people with homogeneously coloured regions with an unchanging background.

In Section 3.4 we describe a fast method for tracking multi-coloured objects using Gaussian mixture models by filtering pixels on the basis of conditional probabilities and estimating moments of the resulting sample set. It functions in real-time on extremely modest hardware and does not require constructing or maintaining three-dimensional models of the object and which is able to cope with moving backgrounds. In Section 3.5 we extend this to a Bayesian framework which performs filtering by classifying pixels based on colour models of both foreground and background.

3.1.4 Dealing with changing lighting

A drawback with colour is that photometric features are highly sensitive to changing conditions such as the viewing geometry and in particular lighting conditions. The human visual system is largely able to cope under these circumstances through *colour constancy*, which is the ability to perceive a colour (or brightness) as constant despite objective fluctuations in chromaticity and luminance (see e.g. McCann et al. [136], Brainard and Wandell [17], Lucassen [124]). Computer vision approaches to colour constancy attempt to reconstruct the spectral composition of the incident light and adjust observed reflectances accordingly (e.g. Forsyth [59]). However, such methods do not perform adequately in any but the most controlled environments.

In Section 3.6 we address the issue of changing conditions by adopting an adaptive statistical framework that iteratively modifies the internal model to keep pace of changing circumstances. Additionally, the process is *selective* and attempts to detect when the tracker has lost the target (e.g. through occlusion) in order to prevent adaptation to contaminated samples. Once reattached to the target, the adaptation process is permitted to proceed.

3.2 Gaussian mixture models of colour

Although colour histograms can be used to estimate distributions in colour space, the level of quantisation imposed on the colour space influences the resulting distribution. If the number of bins n is too high, the estimated distribution will be "noisy" and many bins will be empty. If n is too low, distribution structure is smoothed away. Histograms are effective only when n can be kept relatively low and where sufficient data are available. A potentially more effective "semi-parametric" approach for colour distribution estimation is to use Gaussian mixture models. In this approach, a number of Gaussian functions are taken as an approximation to a multi-modal distribution in colour space and conditional probabilities are then computed for colour pixels.

Let the conditional distribution for a colour $\mathbf{x}$ from a pixel belonging to a multi-coloured object O be a mixture with M component densities:

$$p(\mathbf{x}|O) = \sum_{j=1}^{M} p(\mathbf{x}|j)P(j) \tag{3.1}$$

where a mixing parameter $P(j)$ corresponds to the prior probability that pixel $\mathbf{x}$ was generated by component j and where $\sum_{j=1}^{M} P(j) = 1$. Each mixture component is a Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, i.e. in the case of a 2D colour space:

$$p(\mathbf{x}|j) = \frac{1}{2\pi|\mathbf{\Sigma}_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\mathbf{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)} \tag{3.2}$$

Expectation-Maximisation (EM) is an effective, widely known and established maximum-likelihood algorithm for fitting such a mixture to a data set [12, 179].

Figure 3.1 shows an example of a Gaussian mixture model of a multi-coloured object in HS-space.

Outlier points, which can be caused by image noise and specular highlights, have little influence upon the mixture model.

The semi-parametric nature of mixture models pertains to the need to choose the number of mixture components M. This is similar to the need to choose the number of bins n for a histogram. However, the resulting mixture model is arguably less sensitive to the value of M chosen than histograms are to the value of n. Further, the benefits of the mixture model approach are:

1. The model is capable of smoothing over gaps in the dataset (which may be quite small)

2. The number of components may intuitively be associated with the number of distinct colours on the object being modelled, with each distinct colour considered as the prime source for each of the peaks in the resulting distribution

3. The use of overlapping Gaussians is a natural way of assigning differently sized and oriented regions of the feature space to different "sources", unlike with histograms which consist of equally-sized and equally-spaced non-overlapping bins

However, it is still desirable to be able to select the number of Gaussian components (model order) automatically. Next, we describe a cross-validation based method for doing this in an incremental fashion.

3.3 Automatic model order selection

Model order selection is the problem of choosing the number of parameters that facilitates the accurate modelling of an underlying distribution for a set of data. In this section, we describe a constructive method for automatic determination of the number of components for a colour mixture model.

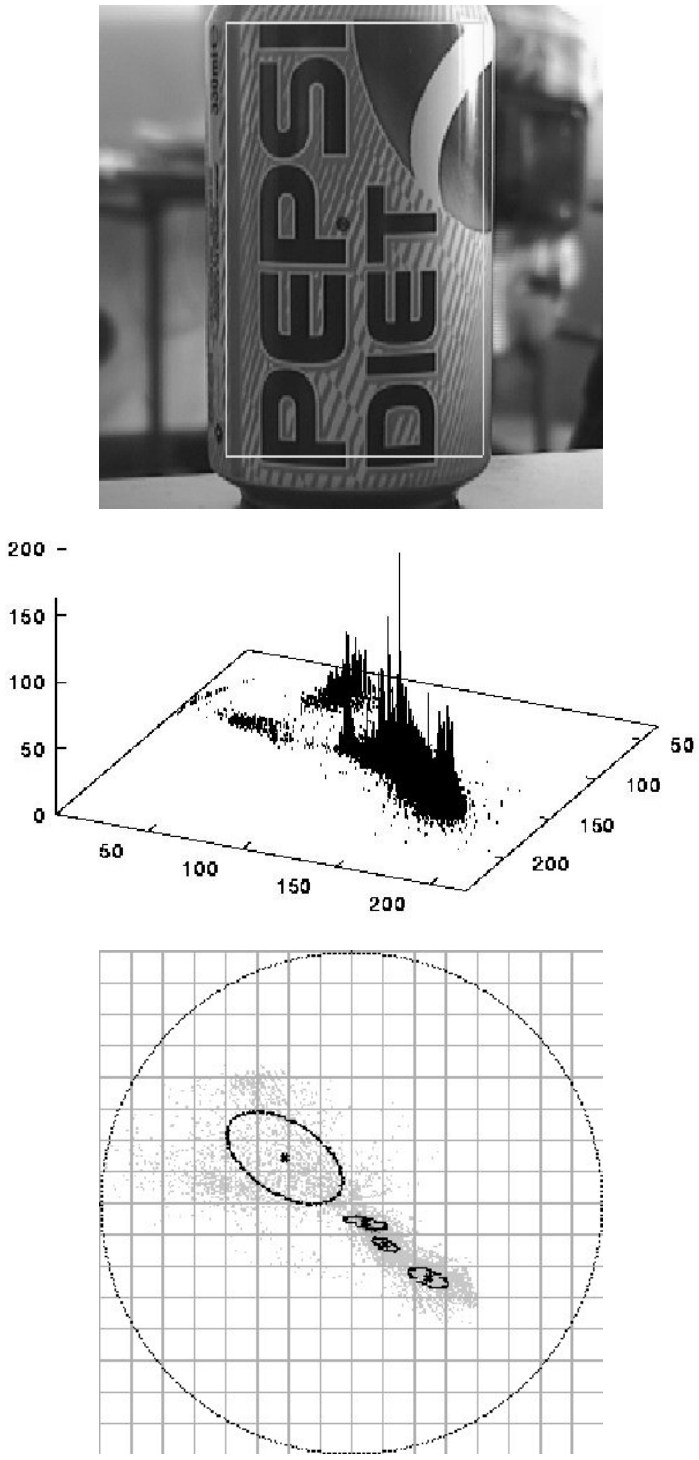


Figure 3.1: An example of modelling the colour distribution of a multi-coloured object in Hue-Saturation space. Top: a multi-coloured object (a drinks can). Middle: its colour histogram (polar coordinates superimposed onto a Cartesian grid). It can be noted that since histograms are non-parametric, such a representation is only viable when a large amount of data is available. Bottom: its Gaussian mixture model. The mixture components are shown as elliptical contours of equal probability.

A standard technique employed for model training, known as *cross validation*, attempts to find the model order that provides the best trade-off between bias and variance. A number of models of different order are trained by minimising an error function for a training set. These models are then evaluated by computing the error function for an independent *validation set*. The model with the lowest error for the validation set is considered to exhibit the best generalisation and its order is taken to be optimal.

This concept is applied to the generation of mixture models through an iterative scheme of splitting components and monitoring generalisation ability. The available data set is partitioned into disjoint training and validation sets. A mixture model $\mathbf{\Phi}$ is typically initialised with a single component (although a larger number may also be used to start with). Model order is then gradually increased by iteratively applying EM and splitting components. The log-likelihood $\mathcal{L}(V;\mathbf{\Phi})$ for the validation set V is computed after every iteration:

$$\mathcal{L}(V;\mathbf{\Phi}) = \sum_k \ln\{p(\mathbf{x}_k|\mathbf{\Phi})\}, \qquad \mathbf{x}_k \in V \tag{3.3}$$

It is assumed that the optimal model order corresponds to the peak in this likelihood function over time. Here, the techniques for selecting and splitting components are outlined and we present a corresponding algorithm called Iterative Model Order Selection (IMOS).

3.3.1 Splitting components

For each component j, let us define a total responsibility r_j on the validation set V as:

$$r_j = \sum_k p(j|\mathbf{x}_k) = \sum_k \frac{p(\mathbf{x}_k|j)P(j)}{\sum_{i=1}^{M} p(\mathbf{x}_k|i)P(i)} \tag{3.4}$$

Then the component κ with the lowest total responsibility for the validation set is selected for splitting:

$$\kappa = \arg\min_j(r_j) \tag{3.5}$$

Once the component κ to be split has been selected, two new components with means $\boldsymbol{\mu}_{\eta 1}$ and $\boldsymbol{\mu}_{\eta 2}$, and covariance matrices $\mathbf{\Sigma}_{\eta 1}$ and $\mathbf{\Sigma}_{\eta 2}$ are computed by:

$$\boldsymbol{\mu}_{\eta 1} = \boldsymbol{\mu}_k + \frac{\lambda_1}{2}\mathbf{u}_1 \tag{3.6}$$

$$\boldsymbol{\mu}_{\eta 2} = \boldsymbol{\mu}_k - \frac{\lambda_1}{2}\mathbf{u}_1 \tag{3.7}$$

$$\boldsymbol{\Sigma}_{\eta 1} = \boldsymbol{\Sigma}_{\eta 2} = \boldsymbol{\Sigma}_k \tag{3.8}$$

where λ_1 is the largest eigenvalue of the covariance matrix $\boldsymbol{\Sigma}_\kappa$ and $\mathbf{u}_1$ is the corresponding eigenvector.

The prior probabilities for the new components are assigned like so:

$$P(\eta 1) = P(\eta 2) = \frac{P(\kappa)}{2} \tag{3.9}$$

3.3.2 A constructive algorithm for model-order selection

Here we describe the Iterative Model Order Selection (IMOS) algorithm. Let M denote the number of components, Φ_M denote the model with M components and $\mathcal{L}(V;\Phi_M)$ the log-likelihood of the validation set with respect to model Φ_M. The initial number of components may be set to a low number (here M is initially set to 1). With a validation set V generated for the generalisation test, a constructive algorithm for model order selection is shown in Algorithm 3.1. The algorithm terminates when a peak is detected in the log-likelihood measurements for the validation set. The application of this algorithm to a data set for a multi-coloured object (the Pepsi can as shown in Figure 3.1) is illustrated in Figure 3.2.

3.4 Fast colour-based tracking

Often, tracking an object requires estimating no more than its position and size in each frame of a sequence. In order to do so, a subset of the total image is essentially identified and the required states estimated from this set. Considering all pixels of an image as a sample population, tracking then amounts to a sampling process involving the filtering of image samples. The corresponding sampling strategy is then the evaluation and acceptance or rejection of samples for inclusion in the desired sample set.

In each frame of a sequence, the sampling strategy conducts two filtering steps. Firstly, samples are filtered on the basis of their position relative to a restricted search window defined by the expected size and position of the object, with those lying outside immediately rejected. Secondly, a confidence map is computed for the image within the search window, which involves

Input: Training data T, validation set V

Output: Gaussian mixture model Φ_M with M components

Set number of components $M = 1$;

Compute mean and covariance matrix for T to initialise Φ_1;

Compute $\mathcal{L}(V;\Phi_1)$ for Φ_1 on V (Equation 3.3);

repeat

  Find component j with lowest total responsibility for V (Equations 3.4 and 3.5);

  Split component j (Equations 3.6, 3.7 and 3.8);

  Set $M = M + 1$;

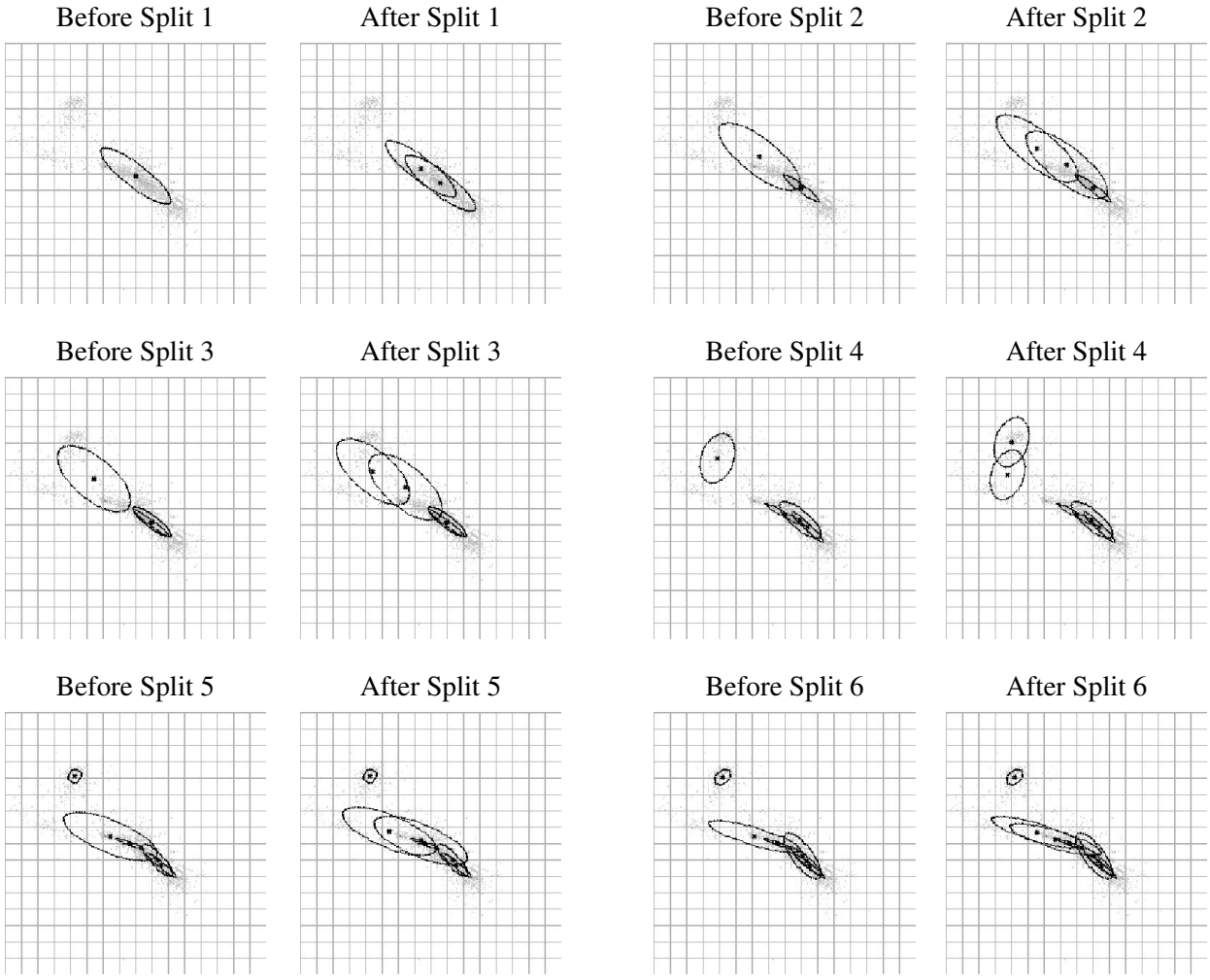
  Apply Expectation-Maximisation for new model Φ_M on T;

  Compute $\mathcal{L}(V;\Phi_M)$ for Φ_M on V;

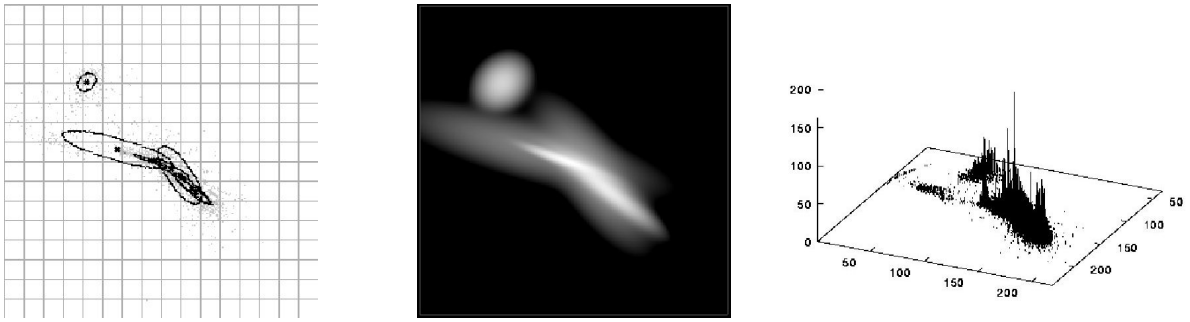
until $\mathcal{L}(V;\Phi_M) < \mathcal{L}(V;\Phi_{M-1})$;

Return Φ_{M-1};

Algorithm 3.1: The Iterative Model Order Selection (IMOS) algorithm for Gaussian mixture models. The model is initialised with a single component. Thereafter, the algorithm repeatedly splits the component with the lowest responsibility for the validation set (Equations 3.4 and 3.5) to create a new higher-order model, applies Expectation-Maximisation to fit the new model to the training set and monitors the log-likelihood of the validation set with respect to the new model. At this stage, a lower log-likelihood than the previous one is considered as an indication of overfitting and the previous model (corresponding to a peak in the likelihood progression) returned as the final result.



(a)



(b)

Figure 3.2: Application of the IMOS algorithm for generating a Gaussian mixture for the colours of the drinks can shown in Figure 3.1. Part (a) illustrates six iterations of the process, with each pair of images showing EM convergence followed by the splitting of a component. Part (b) shows the final seven component model (left) and the resulting probability density function with brighter regions corresponding to higher probabilities (middle). Finally, a histogram of the training data in polar coordinates is shown superimposed onto a Cartesian grid (right).

evaluating the colours for each sample inside given a model O of the colours of the object. The size and position of the object being tracked are then estimated from the resulting distribution in the image plane. Object position at frame t is taken to be the mean $\mathbf{m}_t = (m_x, m_y)$ and object size is estimated from the standard deviation $\boldsymbol{\sigma}_t = (\sigma_x, \sigma_y)$. More precisely, for a given frame t, the object position $\mathbf{m}_t$ is estimated as an offset from the position estimate for the previous frame, $\mathbf{m}_{t-1}$:

$$\mathbf{m}_t = \mathbf{m}_{t-1} + \frac{\sum_{\boldsymbol{\xi}} p(\mathbf{x}_{\boldsymbol{\xi}}|O)(\boldsymbol{\xi} - \mathbf{m}_{t-1})}{\sum_{\boldsymbol{\xi}} p(\mathbf{x}_{\boldsymbol{\xi}}|O)} \tag{3.10}$$

where $\boldsymbol{\xi}$ ranges over all image coordinates in the region of interest, $\mathbf{x}_{\boldsymbol{\xi}}$ is the colour feature vector at pixel coordinate $\boldsymbol{\xi}$ and O is the object model.

The size of the object is estimated by computing the standard deviation of the normalised image probability distribution:

$$\boldsymbol{\sigma}_t = \sqrt{\frac{\sum_{\boldsymbol{\xi}} p(\mathbf{x}_{\boldsymbol{\xi}}|O)\{(\boldsymbol{\xi} - \mathbf{m}_{t-1}) - \mathbf{m}_t\}^2}{\sum_{\boldsymbol{\xi}} p(\mathbf{x}_{\boldsymbol{\xi}}|O)}} \tag{3.11}$$

3.5 Modelling colour in environmental context

When tracking with a single foreground model, it is often of practical necessity to threshold the conditional probabilities $p(\mathbf{x}_{\boldsymbol{\xi}}|O)$ computed for samples. In doing so, values lower than the threshold are taken to be background and are consequently set to zero in order to nullify their influence on the estimation of $\mathbf{m}_t$ and $\boldsymbol{\sigma}_t$. Thresholding probabilities generated by a foreground model alone is often ineffective due to severe overlap between background and foreground colour distributions. For dealing with multi-coloured objects in dynamic scenes, it is desirable to model the colour distribution of the background scene in addition to the objects to be tracked. Doing so also adds further rigour, both intuitively and computationally, to the sampling strategy and improves confidence in the filtered samples. Given a model (such as a Gaussian mixture density) for both an object, O, and the background scene, B, the probability that the colour value $\mathbf{x}_{\boldsymbol{\xi}}$ for a given pixel $\boldsymbol{\xi}$ belongs to the object is given by the posterior probability $P(O|\mathbf{x}_{\boldsymbol{\xi}})$:

$$P(O|\mathbf{x}_{\boldsymbol{\xi}}) = \frac{p(\mathbf{x}_{\boldsymbol{\xi}}|O)P(O)}{p(\mathbf{x}_{\boldsymbol{\xi}}|O)P(O) + p(\mathbf{x}_{\boldsymbol{\xi}}|B)P(B)} \tag{3.12}$$

with the probability it belongs to the background given as $P(B|\mathbf{x}_{\boldsymbol{\xi}}) = 1 - P(O|\mathbf{x}_{\boldsymbol{\xi}})$. The prior probability, $P(O)$, may be set to reflect the expected size of the object within the search area of

the scene $[P(B) = 1 - P(O)]$. Pixels can be assigned according to the Maximum A-Posteriori (MAP) estimate, i.e. to the class that maximises the posterior probability. In this case, since there are only two classes, the class to assign is object O if $P(O|\mathbf{x}) > 0.5$ (Equation 3.12) or background scene B otherwise.

This minimises the probability of misclassification error in a Bayesian sense. However, it is preferable to use the posterior probabilities directly in order to estimate the spatial extent of the object. Furthermore, the density estimates provide a measure of confidence. Pixels in areas of colour space where both foreground and background likelihoods are low are classified with low confidence.

Modelling foreground and background separately also has the practical advantage that the object and scene data can be acquired independently. A single background scene model can subsequently be used with many different objects. This is useful for tracking multiple objects within the same environment. By the same token, a single object model may be used to track the same object in multiple environments, with the caveat that the different environments are similarly illuminated.

3.6 Coping with change

Colour appearance is often unstable due to changes in both background and foreground lighting. The colour constancy problem has been addressed mainly through the formulation of physics-based models (e.g. Forsyth [59]) which attempt to normalise chromatic fluctuations by recovering the spectral composition of the illuminant. However, these methods are generally inadequate in most situations. Apart from changes in lighting, colour appearance also varies over time due to changes in viewing geometry and changes in camera parameters such as auto-iris adjustment.

Given these dynamic characteristics of uncontrolled environments, visual sampling strategies need to be flexible and adaptive. In the context of the framework described here, the pixel filtering process depends on the relevance of the colour models that underpin and guide the process. Under the assumption that viewing conditions change gradually over time, statistical colour models can be adapted to reflect the changing colour appearance of a tracked object (or the background scene against which it is tracked). In this section, we describe a method for adapting Gaussian mixture colour models over time.

3.6.1 On-line model adaptation

At each frame, t, a new set of K pixels, $X^{(t)} = \{\mathbf{x}_1, ..., \mathbf{x}_K\}$, is sampled from the object and used to update the mixture model[1]. These new data sample a slowly varying non-stationary signal. Let $\psi^{(t)}$ denote the sum of the posterior probabilities of the data in frame t, $\psi^{(t)} = \sum_{\mathbf{x} \in X^{(t)}} p(j|\mathbf{x})$. The parameters are first estimated for each mixture component, j, using only the new data, $X^{(t)}$, from frame t:

$$\boldsymbol{\mu}^{(t)} = \frac{1}{\psi^{(t)}} \sum_{\mathbf{x} \in X^{(t)}} p(j|\mathbf{x})\mathbf{x}, \qquad P^{(t)}(j) = \frac{\psi^{(t)}}{N^{(t)}} \tag{3.13}$$

$$\boldsymbol{\Sigma}^{(t)} = \frac{1}{\psi^{(t)}} \sum_{\mathbf{x} \in X^{(t)}} p(j|\mathbf{x})(\mathbf{x} - \boldsymbol{\mu}_{t-1})^T(\mathbf{x} - \boldsymbol{\mu}_{t-1}) \tag{3.14}$$

where $N^{(t)}$ denotes the number of pixels in the new data set and all summations are over the data $\mathbf{x}_k \in X^{(t)}$. The mixture model components then have their parameters updated using weighted sums of the previous recursive estimates, $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, P_{t-1}(j))$, estimates based on the new data, $(\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, P^{(t)}(j))$ and estimates based on the old data, $(\boldsymbol{\mu}^{(t-L-1)}, \boldsymbol{\Sigma}^{(t-L-1)}, P^{(t-L-1)}(j))$:

The mean and covariance matrix are updated as follows:

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \frac{\psi^{(t)}}{D_t}(\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}_{t-1}) - \frac{\psi^{(t-L-1)}}{D_t}(\boldsymbol{\mu}^{(t-L-1)} - \boldsymbol{\mu}_{t-1}) \tag{3.15}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1} + \frac{\psi^{(t)}}{D_t}(\boldsymbol{\Sigma}^{(t)} - \boldsymbol{\Sigma}_{t-1}) - \frac{\psi^{(t-L-1)}}{D_t}(\boldsymbol{\Sigma}^{(t-L-1)} - \boldsymbol{\Sigma}_{t-1}) \tag{3.16}$$

$$P_t(j) = P_{t-1}(j) + \frac{N^{(t)}}{\sum_{\tau=t-L}^{t} N^{(\tau)}} \left(P^{(t)}(j) - P_{t-1}(j) \right) - \frac{N^{(t-L-1)}}{\sum_{\tau=t-L}^{t} N^{(\tau)}} \left(P^{(t-L-1)}(j) - P_{t-1}(j) \right)$$

where $D_t = \sum_{\tau=t-L}^{t} \psi^{(\tau)}$. See Appendix A for the derivation. The following approximations are used for efficiency:

$$\psi^{(t-L-1)} \approx \frac{D_{t-1}}{L+1} \tag{3.17}$$

$$D_t \approx (1 - 1/(L+1))D_{t-1} + \psi^{(t)} \tag{3.18}$$

[1] Throughout this thesis, superscript $^{(t)}$ denotes a quantity based only on data from frame t. Subscripts denote recursive estimates.

The parameter L controls the adaptivity of the model. Setting $L = t$ and ignoring terms based on frame $t - L - 1$ gives a stochastic algorithm for estimating a Gaussian mixture for a stationary signal [12, 223].

During the processing of a sequence, new samples of data for adaptation are gathered from a region of appropriate aspect ratio centred on the estimated object centroid under the assumption that these data form a representative sample of the objects' colours. This will hold for a large class of objects. This approach may be improved by using all pixels determined as belonging to the object, especially if using Bayesian posterior computations given both a foreground and background model (Section 3.5). This also enables background-classified pixels to be used to adapt the background model.

3.6.2 Selective adaptation

An obvious problem with adapting a colour model over time is the lack of ground-truth. Any colour-based tracker can lose the object it is tracking due, for example, to occlusion. If such errors go undetected the colour model will adapt to image regions which do not correspond to the object (*model drift*) since the samples used for adaptation will be contaminated by non-object samples. This is clearly undesirable. In order to help alleviate this problem, observed log-likelihood measurements may be used to detect erroneous frames. Adaptation of the model may then be suspended for these frames, giving the tracker the opportunity to recapture the object and resume adaptation.

The adaptive mixture model seeks to maximise the log-likelihood of the colour data over time. Given an object model O, the normalised log-likelihood $\mathcal{L}(X^{(t)};O)$ of the data $X^{(t)} = \{\mathbf{x}_1, ..., \mathbf{x}_K\}$ observed from the object at time t is given by:

$$\mathcal{L}(X^{(t)};O) = \frac{1}{N^{(t)}} \sum_{\mathbf{x}_k \in X^{(t)}} \ln p(\mathbf{x}_k|O) \tag{3.19}$$

At each time frame, $\mathcal{L}(X^{(t)};O)$ is evaluated. If the tracker loses the object there is often a sudden, large drop in the value of $\mathcal{L}(X^{(t)};O)$. This provides a way to detect tracker failure. Adaptation is suspended when such an error is detected. The tracker is then re-bootstrapped by increasing the search space to the maximum size. Adaptation is re-activated when the object is again tracked with sufficiently high likelihood.

A temporal median filter is used to compute a threshold, T for the log-likelihood. Adaptation

is then only performed when $\mathcal{L}(X^{(t)};O) > T$. More specifically, the median, ν, and standard deviation, σ, of $\mathcal{L}(X^{(t)};O)$ are computed for the n most recent above-threshold frames, where $n \leq L$. The threshold T is then set to $T = \nu - k\sigma$, where k is a constant.

3.7 Experiments

In the following we describe a set of experiments in which colour mixture models were applied to object tracking in dynamic scenes. All the experiments ran in real-time (15-20Hz) on an extremely low-specification 200MHz Pentium PC with a Matrox Meteor board.

Most colour cameras provide an RGB (red, green, blue) signal. In order to model objects' colour distributions, the RGB signal is first transformed to make the intensity or brightness explicit so that it can be discarded in order to obtain a high level of invariance to the intensity of ambient illumination. Here the HSV (hue, saturation, value) representation was used and colour distributions were modelled in the 2D hue-saturation space as polar coordinates. Hue corresponds to our intuitive notion of 'colour' and is defined as an angle about the origin of HS space, whilst saturation corresponds to our idea of 'vividness' or 'purity' of colour and is defined as distance from the origin. At low saturation, measurements of hue become unreliable and are discarded. Likewise, pixels with very high intensity are discarded. It should be noted that the HSV system does not relate well to human vision. In particular, the usual definition of intensity as $max(R+G+B)$ is at odds with our perception of intensity. However, this is not important for the tracking application described here. If in other applications it was deemed desirable to relate the colour models to human perception then perceptually-based systems like CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$ could be used instead of HSV.

3.7.1 Fast colour-based tracking

Figure 3.7.1 shows samples from one continuous sequence where a skin colour mixture model was used on an active camera with pan, tilt and zoom capabilities for tracking a face with occlusion, lighting and scale changes. It is clear that the model copes well with the changes in object appearance. Here the colour mixture model is relatively simple in the sense that the object of interest has almost a uniform colour, enabling the number of components to be easily determined.

Figure 3.3: A face is tracked against a cluttered background by an active camera which pans, tilts and zooms.

3.7.2 Modelling colour in environmental context

Model selection becomes more difficult with multi-coloured objects. Figure 3.4 illustrates the multi-coloured object foreground and background models in HS colour space. These resulted from running the IMOS algorithm for automatic model selection (see Section 3.3.2). A context-dependent object model can be given by a combined posterior density (shown in the bottom right) which defines decision boundaries between object foreground and scene background, even when significant overlap exists between the object and the background.

Figure 3.5 shows an application of the context-dependent object (person) model in tracking. Pixels in the scene were classified as person or background using Equation 3.12 with the prior probabilities set to $P(B) = P(O) = 0.5$. Background classified pixels were replaced by an alternative background to illustrate the sample (pixel) filtering process; in other words, the person being tracked was superimposed onto an alternative dynamic scene. The results are surprisingly good for individual pixel classification alone, although imperfect. However, this implies the utility of using other features in conjunction with colour in order to further empower the filtering step, such as an effectively applied combination of photometric and geometric features.

3.7.3 Coping with change

Results shown in Figures 3.6 and 3.7 illustrate the advantage in using an adaptive model. In this sequence the illumination conditions coupled with the camera's auto-iris mechanism resulted in large changes in the apparent colour of the object of interest (the face of a person) as it approached the window. Towards the end of the sequence, the face became very dark, making hue and saturation measurements unreliable. In Figure 3.6, a non-adaptive model was estimated based

Figure 3.4: Colour mixture models of a multi-coloured object (person model) and the context (scene model). The first row shows the data used to build the foreground (person) and the background (laboratory) models. The second row illustrates the probability density estimated from mixture models for the object foreground and scene background. The rightmost image is the combined posterior density in the HS colour space. Here the "bright" regions represent foreground whilst the "dark" regions give the background. The "grey" areas are regions of uncertainty.

Figure 3.5: Illustration of foreground-background classification-based pixel filtering for tracking. The top row outlines the tracked region for segmentation and the second row illustrates superimposition onto an alternative sequence.

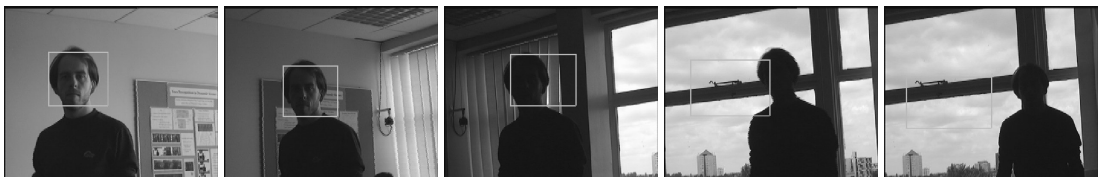


Figure 3.6: Five frames from a sequence in which a face was tracked using a non-adaptive model. The apparent colour of the face changes due to (i) varying illumination and (ii) the camera's auto-iris mechanism which adjusts to bright exterior light.

on the first image of the sequence only and was used throughout. It was unable to cope with the varying conditions and failure eventually occurred. In Figure 3.7, the model was allowed to adapt and successfully maintained a lock on the face.

The experiment shown in Figure 3.8 illustrates the advantage of selective adaptation. The person moved through challenging tracking conditions, before approaching the camera at close range (frames 50-60). Since the camera was placed in the doorway of another room with its own lighting conditions, the person's face underwent a large, sudden and temporary change in apparent colour. When adaptation was performed in every frame, this sudden change had a drastic effect on the model and ultimately led the tracker to fail when the person receded into the corridor. With selective adaptation, these sudden changes were treated as outliers and adaptation was suspended, permitting the tracker to recover.

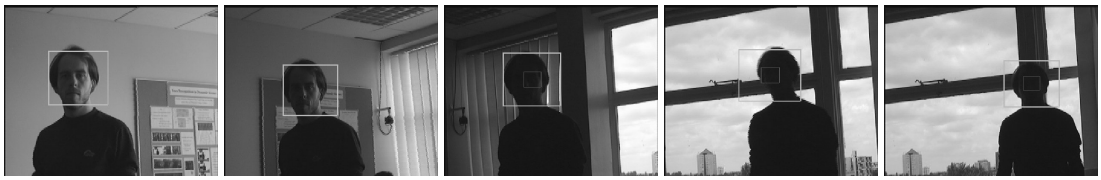


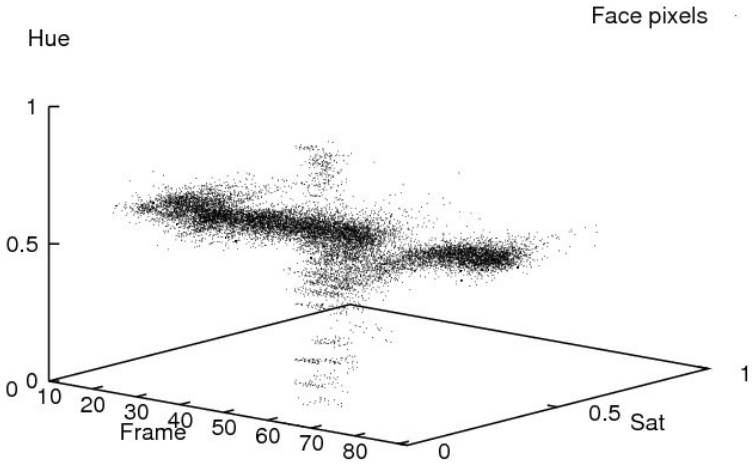
Figure 3.7: The sequence depicted in *Figure 3.6* tracked with an adaptive colour model. Here, the model adapts to cope with the change in apparent colour.

3.8 Discussion

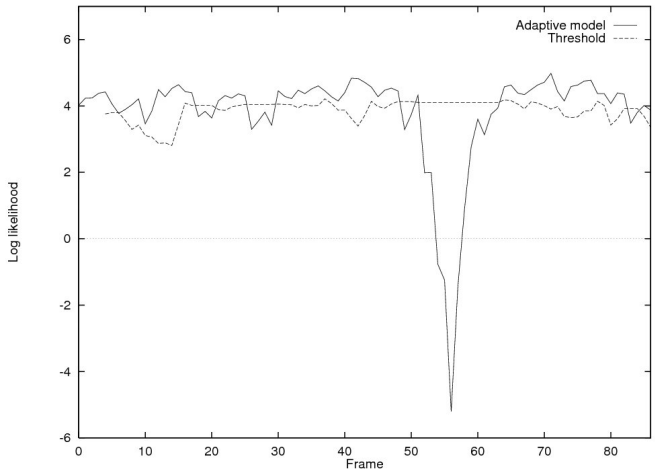
Using a single foreground colour model, fast tracking can be achieved with good robustness, especially when combined with an adaptive model. Furthermore, in some situations, such as tracking a distinctly coloured object, it is able to cope well with moving cameras and changing backgrounds. However, this very fluidity also increases the chance that background and object will occasionally overlap in terms of their characteristics with respect to the model. Background objects that suddenly come into view containing similar colours to the object being tracked may confuse the tracker and cause failure. Modelling the background helps to alleviate this problem, since such shared colours tend to be weighted lower during the sample filtering process, with the result that their influence on state estimates are reduced. This implies the need to adapt background models similarly as for the tracked object. Furthermore, the colour models used here do not capture spatial structure, which has the advantage of enabling greater flexibility for tracking under geometric transformations of both object and scene. The drawback is that geometrical structural differences between object and distractors cannot be exploited for discrimination. Methods employing geometry are hence burdened with the task of estimating geometric dynamics such as the movement of the camera relative to the scene in order to compensate (e.g. [143, 95, 23, 213]). In Chapter 4, we explore the use of different feature types to help cope with the problem of transient relevance encountered with individual feature types by extending the sampling strategy from the spatial domain to the featural domain (Figure 1.8) and building on previous work in this area.

The Iterative Model Order Selection algorithm is a simple and intuitively appealing approach to selecting model order. However, given the dynamic nature of tracking and acknowledged by the use of adaptive models, it is clear that, apart from the actual distributions over the feature space, optimal model order itself is likely to change over time. For example, the number of

(a)



(b)



(c)

(d)

Figure 3.8: (a) Frames 35, 45, 55, 65 and 75 from a sequence. There is strong directional and exterior illumination. The walls have a fleshy tone. At around frame 55, the subject rapidly approaches the camera which is situated in a doorway, resulting in rapid changes in illumination, scale and auto-iris parameters. This can be seen in plot (b) which shows the 3D plot of the hue-saturation distribution over time. In (a), the model was allowed to adapt in every frame, resulting in failure at around frame 60. Plot (b) illustrates the use of selective adaptation. Plot (c) shows the normalised log-likelihood measurements and the adaptation threshold.

distinct colours of an object, each of which produces a peak in the corresponding distribution in feature space, may change with object pose. Consequently, it is desirable to further improve adaptation methodology to cater for this requirement. This would necessarily involve procedures for not only adding components but also removing them when appropriate.

3.9 Summary

In this chapter, we have described a framework for an adaptive sampling strategy for fast, robust object tracking using colour features. The sampling strategy amounts to a filtering process on image samples (pixels) and involves two steps: (1) a focus-of-attention through the use of a search window reflecting the expected position and size of the object; and (2) the evaluation of samples by computing probabilities that they belong to the object. Moments of the probabilities in the image plane are computed to estimate object position and size.

The framework comprises five components:

1. Gaussian mixture models for semi-parametric modelling of the colour distributions of multi-colour objects;

2. An Iterative Model Order Selection (IMOS) algorithm that uses cross-validation for automatically determining the number of components for a Gaussian mixture given a sample set of object colours, a procedure normally performed in an *ad hoc* manner;

3. A sampling strategy for performing fast tracking using colour models;

4. A Bayesian formulation enabling models of object and the environment to be employed together in filtering samples by discrimination;

5. An adaptive mechanism to enable colour models to cope with changing conditions and permit more robust tracking. Furthermore, a method for detecting tracking errors controls adaptation to prevent model drift.

We conducted experiments accordingly:

1. The colour mixture models were used to perform robust object detection and tracking in real-time using extremely modest hardware ;

2. The use of separate colour models for foreground objects and the background scene was described. Successful tracking was thus performed even when there was significant overlap between object and background colour distributions;

3. Selective adaptation of models was shown to improve tracking performance under large changes in apparent colour whilst correctly detecting tracking errors and controlling adaptation.

In the next chapter, we extend the idea of tracking as an adaptive sampling strategy by including the *featural* domain of the Spatial-Featural-Volume (Figure 1.8) in addition to the spatial domain. This is an intuitive step forward since the relevance of different features type in discrimination is in general as fluid as the appearance of an object during a tracking task in difficult environments.

Chapter 4

Multi-Feature Sampling for Tracking

In the previous chapter, we described a framework for an adaptive sampling strategy for tracking using colour features. Such a strategy involves two sequential filtering steps: (a) the restriction of focus of attention to a search window dictated by expectations of object position and size; and (b) the evaluation of samples within the search window with respect to an internal model and a corresponding rejection or acceptance based on the evaluation. Adaptivity is provided by a mechanism for the selective updates of internal models, thereby helping to prevent model-drift and the consequential deterioration in the accuracy of sample evaluations. In this scheme, the sampling process is contained within the spatial domain since samples are formed from a single feature and indexed according to their pixel coordinates.

As described in the previous chapter, object association tasks and, perhaps more acutely, object tracking schemes must cope with imaging fluctuations caused by illumination change, partial or complete occlusions and ever-changing backgrounds caused by moving cameras or distractors in a multi-object environment. Tracking in a cluttered scene under unstable lighting is hindered by the intrinsic instability and transience of features as a useful discriminator for the target object. Furthermore, the robustness of different types of features is highly context-dependent, with photometric and geometric environmental conditions inducing different complications. For example, colour is sensitive to changes in lighting whilst shape and texture may be drastically altered during pose transitions. Additionally, in cluttered scenes dynamic distractors can significantly affect the relevance of specific features over time, e.g. colour may perform adequately when a red target object is tracked against a non-red background but shape may be more discriminative when the

target moves into a red-coloured area. Consequently, in the absence of truly robust features, a successful tracker will not only take measures to alleviate the problems of model-drift but also to utilise the features most likely to be useful at any given time.

In this chapter, we extend the sampling strategy for tracking described in Chapter 3 by incorporating a *third* filtering step to address the issue of selecting appropriate features during tracking. Whilst previously filtering was performed purely in the spatial domain, here the notion is extended to another dimension, the *featural* domain of the Spatial-Featural Volume (SFV) as illustrated in Figure 1.8. This extra dimension constitutes an extra source of pixel samples, with each spatial image coordinate now indexing a pool of values derived from various transformations of the raw image colours. Previous work has addressed the issue of selection from this pool for tracking (e.g. [29, 4, 115, 72]) with the following limitations: (a) in choosing the most relevant features in each frame, features are ranked using metrics or boosting methods which do not address issues of redundancy amongst those selected (see Section 2.1.2); and (b) model drift is inadequately addressed by using static reference models based on unrealistic assumptions of long-term relevance. Here we address these problems within a framework called *Adaptive Multi-Feature Association* (AMA) consisting of two components; (a) A more reliable feature ranking method called *Attribute-based Feature Ranking* (AFR) which consists of a combination of two computed attributes per feature to reduce redundancy and (b) a mechanism called *Multiple Selectively-adaptive Feature Models* (MSFM) for maintaining multiple longer-term reference models that are selectively adapted on-line to avoid model drift. This forms an extension of the dynamic sampling strategy concept to multiple feature domains.

4.1 Scope of the problem

The problem of tracking objects with changing appearance against the background has been addressed by treating tracking as a binary classification problem [119, 148, 33, 29, 4, 115]. Classification has also been used to disambiguate difficult situations in crowded scenes where the close proximity of objects may confuse a tracker [204]. In particular, for coping with constantly varying foreground and background appearance, a feature selection approach has increasingly been employed as a means to selecting the most discriminative appearance characteristics for performing object association given the conditions at any specific time. In this vein, Collins et al. [29] proposed a framework involving the continuous ranking of individual feature types

from a pool over the duration of a tracking task and the use of high-ranked features to generate classifiers for pixel classification in subsequent frames. Periodically, foreground and background pixel samples collected from an object-centred centre-surround bounding box are used to evaluate the discriminative power for each of the features using the variance ratio of their statistics to determine the N most discriminative features. Classifiers are then created for these feature types and used to generate confidence maps to which mean-shift [32] processes are then applied and combined. This idea was extended further by Avidan [4] using AdaBoost [61] to generate and maintain an ensemble of weak classifiers which may implicitly weight different features in a pool rather than perform selection. The weak classifiers are combined into a strong classifier used for generating confidence maps. Samples are collected for each frame and used to generate new weak classifiers in an off-line manner, replacing the weakest classifiers for the current ensemble. Similarly, Grabner et al. [71] proposed an on-line boosting algorithm to maintain an ensemble. Liang et. al [115] modified the framework by Collins et al. [29] to use Bayes error rates instead of the variance ratio to deal with multimodal feature distributions. They also included a method to adapt estimates of object scale using a set of simple correlation templates to roughly estimate object boundaries.

In order to avoid model drift over time, Collins et al. [29] and Liang et al. [115] incorporate a static reference model of target object appearance. This involves combining pixel samples from the first frame of a tracking task with frame-specific pixel samples when ranking. For similar effect, Avidan [4] proposed exempting replacement of the strongest classifier from the first frame of tracking. Grabner et al. [72] took the more sophisticated approach of employing a semi-supervised on-line boosting approach based on the SemiBoost algorithm (Mallapragada et al. [132]). This involves the use of a fixed "prior" classifier acting as a reference model and a separate strong classifier. In each frame, the strong classifier is used to generate confidence maps and estimate object state, after which randomly-selected unlabelled image patches are extracted and classified using a combination of both the strong and prior classifiers. The classes and corresponding weights of the image patches are then used to update the strong classifier. This approach affords the tracker greater flexibility to adapt without allowing it to lose relevance; however, the prior classifier is trained on initial labelled examples and remains static throughout the tracking process.

In this chapter, we address two fundamental limitations of all of these previous methods.

Firstly, the schemes by Collins et al. [29] (using variance ratio) and Liang et al. [115] (using Bayes error) rank features in isolation which can result in highly redundant feature sets giving significantly less discriminative representations of target objects. Boosting approaches, such as those employed by Avidan [4] and Grabner et al. [70, 71, 72], can also suffer from the same problem [55]. Secondly, the attempts at addressing model drift by Collins et al. [29], Liang et al. [115], Avidan [4] and Grabner et al. [72] all inherently either directly or indirectly assume the long-term relevance of target object appearance at initialisation, either by incorporating samples directly from the initial frame or using classifiers trained on a static set of initial samples. While these approaches may be effective in relatively stable environments where very short-term changes may take place, in situations where target object appearance undergoes long-term change, for example due to lighting, the maintained reference samples or classifier quickly become irrelevant. Both limitations contribute to inaccurate pixel classification which in turn results in lower quality pixel confidence maps and more tracking errors.

To address these problems, we argue that combining knowledge of trends in target object appearance over time with object appearance on a frame-specific basis is a more natural approach to overcoming the problems caused by both short-term complications such as occlusion as well as longer-term issues such as lighting change. A key factor, and one neglected by previous methods, is that the feature selection approach to tracking requires a balance between rigidity and flexibility in feature selection. Too much rigidity and the tracker cannot adapt to new situations, too much flexibility and it becomes too susceptible to noise and model drift. Trends should be monitored over time to facilitate *dynamic* reference models as opposed to static. Doing so adds a degree of flexibility to the overall sampling strategy for the tracking task and in the process improves the strength and stability of features selected as well as the accuracy of pixel classification from frame to frame. To that end, we propose a novel on-line adaptive feature ranking and selection framework that balances the long-term featural characteristics of an object with short-term discriminative requirements throughout tracking. We call this framework *Adaptive Multi-feature Association* (AMA) which comprises two components. Firstly, we perform object feature selection by ranking each object feature on the basis of two attributes in order to reduce redundancy in the chosen set: (a) discriminability; and (b) independence to other features. We call this *Attribute-based Feature Ranking* (AFR). Secondly, we maintain multiple feature reference models, one for each feature type in a pool and these are selectively adapted over time. We

refer to these as *Multiple Selectively-adaptive Feature Models* (MSFM) which are combined with distributions of current image data for object feature ranking and foreground pixel classification.

The introduction of a feature selection component results in a three-step sample filtering arrangement for the overall tracking-based sampling strategy as opposed to two steps for the tracking framework in Chapter 3:

1. Focus-of-attention to narrow the search;
2. Sampling from the featural dimension of the Spatial-Featural Volume (Figure 1.8);
3. Sampling from the spatial dimension of the SFV.

We compare the performance of our approach against the framework proposed by Collins et al. [29], Avidan's Ensemble Tracking [4] and the SemiBoost tracker of Grabner et al. [72] in challenging outdoor and indoor scenarios and demonstrate that our scheme results in more stable feature selection, more accurate pixel classification and significantly more robust tracking under very difficult lighting and viewing conditions.

4.2 Adaptive Multi-feature Association

We define feature selection and pixel classification based on a combination of current evidence and adaptive feature reference models for each feature type. More specifically, in each image frame t and for each feature type X_j, $j = 1..J$ from a feature pool of J features, pixels for foreground F_t and background B_t are gathered from an object centered centre-surround bounding box and corresponding distributions $p(X_j|F_t)$ and $p(X_j|B_t)$ are generated. We employ a feature pool comprising linear combinations of RGB values (more details are given in Section 4.3). A set of J feature reference models $p(X_j|M_t)$, one for each feature type, are generated and maintained adaptively over time using a heuristic update method. A set of priors $P_j(F_t)$, $P_j(B_t)$, $P_j(M_t)$ and $P_j(O_t)$ are also heuristically estimated in each frame. The foreground distribution $p(X_j|F_t)$ at time t is combined with the corresponding feature reference model $p(X_j|M_t)$ to form a model describing a target object $p(X_j|O_t)$. Given $p(X_j|F_t)$ and $p(X_j|B_t)$, each feature type X_j is ranked and the N highest ranked used individually for Bayesian posterior classification of pixels in the subsequent frame using the estimated priors and both $p(X_j|O_t)$ and $p(X_j|B_t)$. For each of the N selected feature types, the confidence measures of those pixels classed as target object are used to generate confidence maps which are normalised. A mean-shift process is applied to the weighted

average of the N maps to estimate target object position. The corresponding N feature reference models are then heuristically updated on the basis of previously observed foreground and background feature characteristics. This framework, which we call Adaptive Multi-feature Association (AMA), consists of two key components as follows: (1) Attribute-based Feature Ranking for selecting feature types with reduced redundancy; and (2) Multiple Selectively-adaptive Feature Models for maintaining a long-term appearance reference for a tracked object. We now describe each of these in turn.

4.2.1 Attribute-based Feature Ranking

Feature selection for foreground pixel classification can be considered as choosing the K features $\{X_{b_1},...,X_{b_K}\}$ from a pool of J features $X_1,...,X_J$ that minimise the conditional entropy $H(Y|X_{b_1},...,X_{b_K})$, where Y is a class variable. Ideally, such a set is as small as possible to reduce descriptive redundancy amongst subsets of such features. However, this is in general computationally intractable. Algorithms exist that attempt to recover the Markov blanket of the class variable (e.g. Yaramakala [241] and more recently Fu and Desmarais [63]); however, these are non-greedy algorithms which can require many estimates of conditional independence to be made at each iteration, increasing computation cost which is undesirable for a time-critical task such as object tracking. The Conditional Mutual Information Maximisation (CMIM) algorithm [55] is a fast, principled suboptimal greedy method for finding such features wherein each feature selected maximises the mutual information with the class variable conditional on the features previously selected. However, this algorithm at each iteration requires the estimation of joint distributions for three random variables, the class variable Y and two features being compared. Tracking small objects in low resolution images typically lacks sufficient data from each frame for a reliable estimate of such joint distributions, which may make the use of such algorithms unreliable.

To address the problem of reducing redundancy, we approximate the choosing of descriptive features by computing two attributes per feature X_j for each frame t which we then average for ranking:

(1) ***Discriminability*** $d_{j,t}$ – Given K sample pixel feature values $\mathbf{x}_j^k$ for feature X_j from frame t, $k = 1..K$, $\mathbf{x}_j^k \in \mathcal{F}_j$ where $\mathcal{F}_j$ is the set of all possible values for feature X_j, we generate distributions $p(X_j|F_t)$ and $p(X_j|B_t)$ for foreground and background respectively and compute the variance

ratio as per Collins et al. in [29]:

$$d_{j,t} = \frac{var\left\{j,t,\frac{1}{2}\left[p(\cdot|F_t)+p(\cdot|B_t)\right]\right\}}{var\left\{j,t,p(\cdot|F_t)\right\}+var\left\{j,t,p(\cdot|B_t)\right\}} \tag{4.1}$$

where

$$var\left\{j,t,\Psi(\cdot)\right\} = \left\{\sum_{\mathbf{x}\in\mathcal{F}_j}\Psi(\mathbf{x})\log^2\left(\frac{p(\mathbf{x}|F_t)}{p(\mathbf{x}|B_t)}\right)\right\} - \left\{\sum_{\mathbf{x}\in\mathcal{F}_j}\psi(\mathbf{x})\log\left(\frac{p(\mathbf{x}|F_t)}{p(\mathbf{x}|B_t)}\right)\right\}^2 \tag{4.2}$$

The variance ratio could be replaced by another measure of discriminability such as classification error rate on the samples taken from frame t; however, we use this to more clearly observe the influence of the second attribute (described below) on feature ranking as compared to Collins et al. [29].

(2) ***Independence*** $u_{j,t}$ – We approximate the degree of independence between features by computing one minus the average canonical correlation [86] between the K pixel samples collected for feature j, $\mathbf{x}_j^k \in R^m$ and the corresponding K pixel samples for all other features i from the feature pool, $\mathbf{x}_i^k \in R^n$, $i \neq j$.

We deploy Canonical Correlation Analysis (CCA) to find the basis functions $\langle \mathbf{w}_j, \mathbf{w}_i \rangle$ for which the projections $x_j = \mathbf{w}_j^T\mathbf{x}_j$ and $x_i = \mathbf{w}_i^T\mathbf{x}_i$ are maximally correlated. More specifically, CCA maximises the following:

$$\rho_{j,i} = \frac{E[x_j x_i]}{\sqrt{E[x_j^2]E[x_i^2]}} = \frac{E[\mathbf{w}_j^T\mathbf{x}_j\mathbf{x}_i^T\mathbf{w}_i]}{\sqrt{E[\mathbf{w}_j^T\mathbf{x}_j\mathbf{x}_j^T\mathbf{w}_j]E[\mathbf{w}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{w}_i]}} = \frac{\mathbf{w}_j^T\mathbf{C}_{ji}\mathbf{w}_i}{\sqrt{\mathbf{w}_j^T\mathbf{C}_{jj}\mathbf{w}_j\mathbf{w}_i^T\mathbf{C}_{ii}\mathbf{w}_i}} \tag{4.3}$$

where $\mathbf{C}_{jj}$ and $\mathbf{C}_{ii}$ denote covariance matrices for features j and i respectively and $\mathbf{C}_{ji}$ is the covariance between j and i. A total of $V = min(m,n)$ pairs of basis functions may be obtained $\langle \mathbf{w}_j^v, \mathbf{w}_i^v \rangle$, $v = 1..V$, with corresponding correlations $\rho_{j,i}^v$ by successively solving Equation 4.3 subject to the constraint that $\langle \mathbf{w}_j^v, \mathbf{w}_i^v \rangle$ are orthogonal to all preceding pairs found. We then estimate a measure of independence $u_{j,t}$ as the average over the V dimensions and correlations between feature j and all other features i in the pool:

$$u_{j,t} = 1 - \frac{1}{V(J-1)}\sum_{i\neq j}\sum_{v}\rho_{j,i}^v \tag{4.4}$$

The final feature ranking $r_{j,t}$ is computed as the mean of Discriminability and Independence:

$$r_{j,t} = \frac{1}{2}(d_{j,t} + u_{j,t}) \tag{4.5}$$

By combining these two attributes, the rankings for features considered strongly discriminative by variance ratio are enhanced according to their level of correlation with other features in the pool. Highly-correlating features are considered as embodying redundant information and their ranking value $r_{j,t}$ increased less than those with low average correlation with the other features.

4.2.2 Multiple Selectively-adaptive Feature Models

Rather than maintain a static set of reference pixels for combining with current image data at each frame [29, 115, 4], we consider multiple adaptive feature reference models $p(X_j|M_t)$, one for each feature X_j in the pool, which are heuristically defined in terms of previously observed foreground and background characteristics and adapted during tracking to maintain the most useful target object discriminators within each feature domain. Adaptation is performed only for the N highest ranked features in order to maintain a level of confidence in the data being used for the update. More precisely, $p(X_j|M_t)$ for feature X_j at frame t is heuristically estimated as:

$$p(X_j|M_t) = \frac{1}{\kappa} \max \left\{ 0, \sum_{l=t-L+1}^{t} [P_j(F_l)p(X_j|F_l) - P_j(B_l)p(X_j|B_l)] \right\} \tag{4.6}$$

where κ is a normalisation constant, L is a temporal window controlling the rate of adaptation of feature reference models, l indexes frames within the window, $p(X_j|F_l)$ and $p(X_j|B_l)$ are foreground and background distributions respectively and corresponding priors are $P_j(F_l)$ and $P_j(B_l)$. A smaller value for L enables the model to adapt more strongly to shorter-term changes. With this formulation, a given feature vector is essentially weighted by its foreground representation minus its background representation over the temporal window. Over time, feature vectors which are statistically stronger for the foreground reinforce their representation in the model in proportion to their strength; similarly, weakened ones are more associated with the background.

Estimating priors

The prior probabilities $P_j(F_t)$ and $P_j(B_t)$ are estimated heuristically to reflect *confidence* in the descriptive accuracy of the foreground and background sample distributions respectively for feature X_j in frame t. These are important for updating the feature reference models (Equation 4.6) to reduce the chance of model drift over time. In updating these priors, Bayes error estimates may be used to derive two heuristics for distributions of foreground and background data; *stability*, indicating the degree of change in the distribution between frames, and *unreliability*, relating to an overlap between foreground and background. These are then combined to derive a final estimate for the prior. More precisely, the foreground and background priors $P_j(F_t)$ and $P_j(B_t)$

respectively for feature X_j at frame t are computed like so:

$$P_j(F_t) = \frac{1}{2}\left(1 + \int_{X_j} \min\{p(X_j|F_t), p(X_j|F_{t-1})\}\, dX_j - \int_{X_j} \min\{p(X_j|F_t), p(X_j|B_{t-1})\}\, dX_j\right) \tag{4.7}$$

$$P_j(B_t) = \frac{1}{2}\left(1 + \int_{X_j} \min\{p(X_j|B_t), p(X_j|B_{t-1})\}\, dX_j - \int_{X_j} \min\{p(X_j|B_t), p(X_j|F_{t-1})\}\, dX_j\right) \tag{4.8}$$

For each prior, the first integral represents the match between the current and previous distributions for the corresponding class with high values indicating good stability. The second integral is proportional to the degree of overlap between the classes and indicates the level of unreliability of the corresponding class distribution.

Histogram approximation

In practice, we approximate Equations 4.1, 4.2, 4.6, 4.7 and 4.8 by modelling foreground, background and feature reference distributions as histograms $h(\mathbf{x}_j|F_t)$, $h(\mathbf{x}_j|B_t)$ and $h(\mathbf{x}_j|M_t)$ with bins corresponding to discretised values for feature X_j. The histogram for the corresponding reference model is then updated in frame t by:

$$h(\mathbf{x}_j^k|M_t) = \sum_{l=t-L+1}^{t} \max\left\{0, \left[P_j(F_l)h(\mathbf{x}_j^k|F_l) - P_j(B_l)h(\mathbf{x}_j^k|B_l)\right]\right\} \tag{4.9}$$

for all bins k with corresponding feature vector value $\mathbf{x}_j^k$. The priors are estimated using histogram intersection:

$$P_j(F_t) = \frac{1}{2}\left(1 + \sum_k \min\left\{h(\mathbf{x}_j^k|F_t), h(\mathbf{x}_j^k|F_{t-1})\right\} - \sum_k \min\left\{h(\mathbf{x}_j^k|F_t), h(\mathbf{x}_j^k|B_{t-1})\right\}\right) \tag{4.10}$$

$$P_j(B_t) = \frac{1}{2}\left(1 + \sum_k \min\left\{h(\mathbf{x}_j^k|B_t), h(\mathbf{x}_j^k|B_{t-1})\right\} - \sum_k \min\left\{h(\mathbf{x}_j^k|B_t), h(\mathbf{x}_j^k|F_{t-1})\right\}\right) \tag{4.11}$$

The resulting conditional probabilities are derived from the normalised histograms:

$$P(\mathbf{x}_j^k|M_t) = \frac{h(\mathbf{x}_j^k|M_t)}{\sum_i h(\mathbf{x}_j^i|M_t)}, \quad P(\mathbf{x}_j^k|F_t) = \frac{h(\mathbf{x}_j^k|F_t)}{\sum_i h(\mathbf{x}_j^i|F_t)}, \quad P(\mathbf{x}_j^k|B_t) = \frac{h(\mathbf{x}_j^k|B_t)}{\sum_i h(\mathbf{x}_j^i|B_t)} \tag{4.12}$$

Note that the histograms are maintained in their unnormalised state and only normalised during the computation of the conditional probabilities. This is to ensure that the strength of representation for feature values in the feature models $h(\mathbf{x}_j^k|M_t)$ are not capped during the update computation (Equation 4.9).

In summary, we maintain multiple dynamic reference models, one for each feature domain, which are selectively heuristically updated over time using Equation 4.6. These incorporate image evidence over the previous L frames, with foreground and background priors $P_j(F_l)$ and $P_j(B_l)$ from each frame weighting the corresponding evidence. These priors are heuristically estimated according to Equations 4.7 and 4.8, each composed of two Bayes error terms indicating *stability* and *unreliability*. The *stability* term is computed by comparing evidence from the current frame and the previous frame for the same class (e.g. between foreground data for frame t and foreground data from frame $t-1$), indicating the level of change between the distributions. A high Bayes error indicates high stability since change is small. Conversely, a low Bayes error indicates a sudden difference in the distributions which can be caused by tracker localisation error or occlusion, in which case stability is low. The *unreliability* term relates to the match between the statistics for different classes between frames (e.g. between foreground data from frame t and background data from frame $t-1$). A high Bayes error here indicates a strong overlap between the classes and consequently high unreliability. Conversely, a low Bayes error corresponds to low overlap and greater confidence in the data differentiating the two classes. Each prior is then constructed by subtracting the value for unreliability from the value for stability and scaling and shifting the result to ensure a value between zero and one. Low priors directly relate to the level of confidence in the corresponding image data, either as a result of tracking failure or occlusion, with their influence on the reference models consequently reduced.

4.2.3 Tracking by Adaptive Multi-feature Association

As described in Section 4.2.1, there are three filtering steps for the overall sampling strategy for tracking. Firstly, attention is focused on a specific region of the image within which the object is expected to lie. Secondly, features are ranked according to estimates of their discriminability using samples from the previous frame. Finally, a confidence map is generated from a weighted combination of the most highly ranked features by classifying pixels as object or background within the region of focus-of-attention. This confidence map constitutes the final set of samples from which object position may be estimated.

A target-centered approach is deployed to sample foreground and background pixels in each frame for the feature ranking step. The background is assumed to be subject to spontaneous and continuous change as would be expected for situations with moving cameras or busy scenes with multiple moving occluding or proximal distractors. More specifically, an object-centred

box of width w and height h bounds a tracked object. Pixels within are used as foreground feature samples and background appearance is sampled from a region of thickness $\beta \cdot max(w,h)$ surrounding the centre box. The union of pixels from the centre and surround regions are denoted as region R. The determination of this region R facilitates the first filtering step for focus-of-attention in the overall sampling strategy, with pixel samples lying within being subject to further scrutiny and those lying outside being ignored.

In the first frame $t = 1$ of tracking, given an initialised centre box for a target object O in a scene, the feature reference models $p(X_j|M_1)$ for each feature X_j are initialised to null and corresponding foreground and background distributions $p(X_j|F_1)$ and $p(X_j|B_1)$ are generated from the expanded region R. These foreground and background distributions are used for Attribute-based Feature Ranking as described in Section 4.2.1. The feature ranking procedure constitutes the second filtering step of the overall sampling strategy. The best N features are used for the generation of confidence maps in the second frame, which are combined and used for mean-shift estimation of a new box position. In any given frame t, the confidence map $c_{j,t}$ for feature X_j is generated using models from the previous frame by computing the Bayesian posterior $P(O_t|\mathbf{x}_j^k)$ for the feature vector values $\mathbf{x}_j^k$ for all K pixels $k \in R_{t-1}$ of the centre-surround region R_{t-1} estimated for the previous frame:

$$P(O_t|\mathbf{x}_j^k) = \frac{P(\mathbf{x}_j^k|O_{t-1})P_j(O_{t-1})}{P(\mathbf{x}_j^k|O_{t-1})P_j(O_{t-1}) + P(\mathbf{x}_j^k|B_{t-1})P_j(B_{t-1})} \tag{4.13}$$

where the priors for object O_{t-1} and background B_{t-1} sum to unity, $P_j(O_{t-1}) + P_j(B_{t-1}) = 1$ and the conditional probability $P(\mathbf{x}_j^k|O_{t-1})$ is computed as:

$$P(\mathbf{x}_j^k|O_{t-1}) = P(\mathbf{x}_j^k|F_{t-1})P_j(F_{t-1}) + P(\mathbf{x}_j^k|M_{t-1})P_j(M_{t-1}) \tag{4.14}$$

Here we have two new priors, $P_j(M_{t-1})$ and $P_j(O_{t-1})$ for the reference models and object respectively. These can be computed simply as:

$$P_j(M_{t-1}) = 1 - P_j(F_{t-1}) \tag{4.15}$$

$$P_j(O_{t-1}) = 1 - P_j(B_{t-1}) \tag{4.16}$$

from the foreground and background priors for frame $t-1$ (Equations 4.7 and 4.8). A high value for $P_j(F_{t-1})$ results in less emphasis being placed on the reference model M_{t-1} and vice-versa

(Equation 4.14). Consequently, low confidence in current image evidence results in more weight being placed on the reference model.

In practice we use the log-likelihood ratio to classify pixels with the sign denoting class and magnitude the confidence. For the top N features' confidence maps $c_{n,t}(k)$, $n = 1..N$ where k indexes a pixel, we set all negative (background) values to zero and derive the final confidence map $C_t(k)$ as a weighted average of the normalised individual maps:

$$c_{n,t}(k) = max\left\{0, \log\left(\frac{P(\mathbf{x}_n^k|O_{t-1})P_n(O_{t-1})}{P(\mathbf{x}_n^k|B_{t-1})P_n(B_{t-1})}\right)\right\} \tag{4.17}$$

$$C_t(k) = \sum_n \frac{r_{n,t-1}}{\sum_{i=1}^N r_{i,t-1}} \cdot \frac{c_{n,t}(k)}{\sum_{l=1}^K c_{n,t}(l)} \tag{4.18}$$

for the K pixels $k \in R_{t-1}$ where $r_{n,t-1}$ is the ranking score for feature n (Equation 4.5).

The generation of the confidence maps $C_t(k)$ forms the third and final filtering step of the overall sampling strategy. Mean-shift is then applied to $C_t(k)$ to estimate the new target object position and the corresponding N feature reference models updated (Equation 4.9). The new centre-surround box is used to separate new foreground and background samples for the cycle to continue in subsequent frames.

In summary, given an initialised bounding box for the first frame, the bounding box is expanded to yield a surround region and foreground and background distributions $P(X_j|F_1)$ and $P(X_j|B_1)$ generated accordingly from this expanded region. All priors $P_j(F_1)$, $P_j(B_1)$, $P_j(M_1)$ and $P_j(O_1)$ are set to 0.5 and all feature reference models $P(X_j|M_1)$ updated as per Equation 4.6. This then initiates the main tracking cycle which involves:

1. Ranking each feature X_j according to the current foreground and background data $P(X_j|F_t)$ and $P(X_j|B_t)$;
2. Loading the next frame $t+1$;
3. Computing a confidence map for the new frame given the N highest ranked features;
4. Estimating a new bounding box given the current confidence map;
5. Scaling the new bounding box to yield a centre-surround region;
6. Collecting data for new foreground and background distributions $P(X_j|F_{t+1})$ and $P(X_j|B_{t+1})$;

7. Updating the priors $P_j(F_{t+1})$, $P_j(B_{t+1})$, $P_j(M_{t+1})$ and $P_j(O_{t+1})$; and

8. Updating the reference models $P(X_j|M_{t+1})$ for the top N ranked features.

The cycle then repeats. This procedure is detailed in Algorithm 4.1.

4.3 Experiments

We conducted experiments on four scenarios labelled A, B, C and D for comparing the following; (a) the framework presented by Collins et al. [29]; (b) Avidan's Ensemble Tracking method [4]; (c) The SemiBoost tracker by Grabner et al. [72]; and (d) our Adaptive Multi-feature Association (AMA) method. For (d), we show results both for: (1) Attribute-based Feature Ranking (AFR) only, i.e. with static feature reference models in the vein of previous methods; and (2) AFR combined with our Multiple Selectively-adapted Feature Models (MSFM) framework to illustrate how the latter further improves robustness.

4.3.1 Datasets and Settings

Our scenarios consisted of one indoor scene from a concourse of a train station and three outdoor scenes from the PETS 2009 dataset [152]. They are all wide-angle scenes containing multiple moving distractors and low-resolution objects undergoing temporary occlusions. Scenario A is taken from the PETS 2009 database and consists of 40 frames of a crowded scene with complex shadowed and bright regions. This is used to compare the strength and stability of the highest ranked features for each of the methods along with the quality of the resulting confidence maps. Scenario B consists of 250 frames from a low-quality scene of a concourse at a train station and is used to demonstrate the greater robustness and flexibility of AMA in tracking during severe lighting changes. Scenario C is also from PETS 2009 and consists of around 85 frames of a crowd of people moving against high-contrast bright and shadowed regions. This is used to demonstrate the ability of AMA to cope with the temporary severe occlusion of a small target object by other moving objects. Finally, Scenario D is another sequence from PETS 2009 consisting of 160 frames showing a group of people moving to congregate as a crowd in the centre of the scene. In this scenario we demonstrate the robustness of AMA in tracking a small target object undergoing both severe occlusion and severe lighting changes simultaneously. In each scenario, we track a single person as they move through the scene.

Input: Sequence of frames $f_1, ..., f_T$, object bounding box b_1 for frame f_1

Output: Bounding boxes $b_2, ..., b_T$ for frames $f_2, ..., f_T$

Initialise all feature reference models $p(X_j|M_1)$ to null;

Set bounding box $\upsilon = b_1$ with width w and height h;

foreach $t = 2$ *to* T **do**

Expand υ by $\beta \cdot max(w,h)$ for surround region;

Generate $p(X_j|F_t)$ and $p(X_j|B_t)$ for each feature X_j;

if $t = 2$ **then**

Set $P_j(F_1) = P_j(B_1) = P_j(M_1) = P_j(O_1) = 0.5$;

Update $p(X_j|M_1)$ for all features using Equations 4.9 and 4.12;

else

Update $P_j(F_{t-1})$, $P_j(B_{t-1})$, $P_j(M_{t-1})$ and $P_j(O_{t-1})$ using Equations 4.7, 4.8, 4.15 and 4.16;

Update $p(X_j|M_{t-1})$ for top N ranked features using Equations 4.9 and 4.12;

end

Rank each feature X_j using Equations 4.1, 4.4 and 4.5;

Get frame f_t;

Generate confidence map C_t for f_t using Equation 4.17;

Apply mean-shift to C_t for new target position estimate and bounding box b_t;

Set $\upsilon = b_t$;

end

Algorithm 4.1: The Adaptive Multi-feature Association (AMA) algorithm. Tracking initialisation is assumed to be provided along with the first frame. All priors are initialised to 0.5 and these values used for the second frame; they are updated for subsequent frames using Equations 4.7, 4.8, 4.15 and 4.16. Reference models are then updated using Equations 4.9 and 4.12 (all features if the first frame, only the top N if subsequent). All features are then ranked using Equations 4.1, 4.4 and 4.5. The next frame is then collected and confidence maps computed using the log-likelihood ratio (Equation 4.17). Finally, mean-shift is applied to estimate new object position and the cycle restarts.

4.3.2 Implementations

All implementations were done in MATLAB. For all trackers, we employed the same feature pool as Collins et al. [29] which comprised 49 unique linear combinations of R, G and B values with multipliers $\{-2,-1,0,1,2\}$. Values were scaled to between 0 and 31 and distributions modelled as normalised 32-bin histograms.

We implemented the Collins et al. tracker framework largely as described in the original paper [29]; however, since our experiments focus on feature selection and model drift, we did not implement their technique for disambiguating distractors. As per the original work, we retained samples from the first frame (effectively a static reference model) which we combined with evidence in individual frames in order to alleviate model drift. For Avidan's Ensemble Tracking we employed the weak classifiers used in the original paper [4], with the data comprising 49-dimensional feature vectors per pixel, each value representing one of the features from the feature pool. Confidence maps were generated from the dot product between a separating hyperplane and the feature vectors; consequently, the hyperplane acted as a weighting for each of the features in the pool. The strong classifier consisted of five such weak classifiers, with the result that different combinations of weighted features from the pool were used over time. In each frame, the single weakest classifier was replaced by a newly computed one from new sample data. The strongest classifier from the first frame was retained throughout tracking, acting as a static reference model to counteract model drift. The SemiBoost tracker was implemented again according to the method and algorithm given in Grabner et al. [72]. We maintained a strong classifier comprising the strongest weak classifiers from three selectors, with each selector containing 49 weak classifiers, one for each feature from the pool. We employed the same classifiers used for the Collins et al. method as well as for AMA. As per [72], a strong classifier, trained on the first frame, acted as a prior (i.e. a static reference model) to reduce model drift. In all cases, we employed a meanshift operator to update estimates of object position in each frame from the confidence maps generated by each tracker.

For our AMA implementation, we set the background region multiplier β (see Section 4.2.3) to 0.75. The window for updating feature reference models w was set to 50, i.e. sufficiently long (Section 4.2.2). We used the top $N = 3$ ranked features for confidence map generation and model updates, since using more was previously reported to offer little extra for this feature pool [29]. When testing AFR only, we used static feature reference models initialised in the first

frame as per Collins et al. [29] and Liang et al. [115]. Accordingly, we also set all priors for each feature j, $P_j(O)$, $P_j(B)$, $P_j(F)$ and $P_j(M)$ (Section 4.2.1) to a constant 0.5. When testing AFR in combination with MSFM, the priors were updated as per Equations 4.7, 4.8, 4.15 and 4.16.

We now describe in turn the experiments conducted on all four scenarios as described above. For each scenario and for Collins et al., Ensemble Tracking, the SemiBoost tracker and AMA (both AFR only and AFR + MSFM), we show: (a) several output frames with overlaid estimated bounding boxes together with manually labelled ground truth boxes to illustrate object localisation; (b) corresponding pixel confidence maps to show the accuracy of pixel classification; (c) statistics of highest ranked features during tracking to illustrate the level of stability and consistency in feature association; and (d) plots illustrating localisation errors in terms of pixel displacement between estimated bounding boxes and manually-labelled ground truth.

4.3.3 Scenario A: Feature selection performance

The feature selection approach to tracking requires a balance between rigidity and flexibility in feature selection. Too much rigidity and the tracker cannot adapt to new situations, too much flexibility and it becomes too susceptible to noise and model drift. These problems are largely reflected in the quality of the resulting confidence maps from which object positions are estimated. Cleaner and sparser maps are more desirable since they provide fewer distractors for estimating object position. In this experiment, we compare feature ranking statistics and generated confidence maps for each of the trackers.

Figure 4.1 shows the output from each of the trackers for frames 10, 20, 30, 40 and 50 of Scenario A. Each row (a)-(e) illustrates the application of Collins et al., Ensemble Tracking (ET), the SemiBoost tracker (ST), AFR and AFR+MSFM respectively. Each frame shows the output from the tracker with estimated object centre-surround bounding boxes (small yellow and large red boxes respectively). Figure 4.2 shows the corresponding confidence maps generated for the top ranked feature. Although for this example all the trackers maintained a lock on the object successfully (as confirmed by the ground-truth comparison in Figure 4.4), note that with AFR the part of the target most salient in discriminating against distractors was evidently emphasised in the confidence maps (the person's blue top) as compared with Collins et al. and Ensemble Tracking, demonstrating the greater utility of the features selected. The SemiBoost tracker settled to even cleaner confidence maps after Frame 20. However, the maps for AFR+MSFM were consistently the sparsest and most object specific from the beginning and throughout the duration

of the sequence. This is attributed to both a better initial selection of features as well as a tendency of the adaptive models to strengthen the representation of consistently observed feature values over time, which in turn facilitated more stability in future feature selection and accurate pixel classification.

Figure 4.3 shows plots corresponding to the top three ranked (in the case of Ensemble Tracking and the SemiBoost tracker, weighted) features for each frame of tracking from Scenario A. It can be seen that Collins et al. and Ensemble Tracking showed greater fluctuation and instability in their selection of features. This is reflected in the more noisy confidence maps seen in Figures 4.2(a) and 4.2(b). However, AFR was able to overall select more stable and appropriate features during the tracking task. Using MSFM further stabilised the selection process and improved pixel classification. This was reflected in the tighter and cleaner confidence maps from Figures 4.2(d) and 4.2(e).

Although the SemiBoost tracker showed the least variation in highest ranked features, the corresponding confidence maps were not consistently sparse throughout the sequence and showed more noise early on (Figure 4.2(c)), suggesting that those ranked most highly at the beginning were not necessarily the most useful. The feature ranking statistics for the SemiBoost tracker (Figure 4.3) also showed the single same feature being ranked highest from around frame 20 onwards, with the next two best features essentially being discarded by having their weights set to zero. This coincides with the cleaner confidence maps shown in Figure 4.2(c). This turned out to be acceptable for this scenario as the tracker was able to perform successfully; however, as will be shown in the following scenarios it suggests a lack of flexibility which can be detrimental in more challenging tracking situations.

4.3.4 Scenario B: Tracking under severe illumination changes

We now examine tracker performance for a challenging indoor scenario comprising 250 frames undergoing severe lighting change over time. We have two examples of changing illuminant for this scenario; the first showing a change in brightness and the second a change in colour. These changes are simulated by gradually modifying the original pixel values in sequence frames over time. For brightness changes, these modifications are then gradually reversed to original conditions. This allows us to observe the level of robustness of each of the trackers as well as their ability to adapt to such changing conditions.

Figure 4.1: Scenario A Tracker Output: The centre-surround box shows tracker localisation for the object, with the center box denoting estimated object position and the surround region the area from which background samples are gathered. The dashed blue box indicates manually labelled ground truth. The bottom-left of each output frame depicts the tracked region zoomed in for clarity. See Figure 4.2 for corresponding confidence maps.

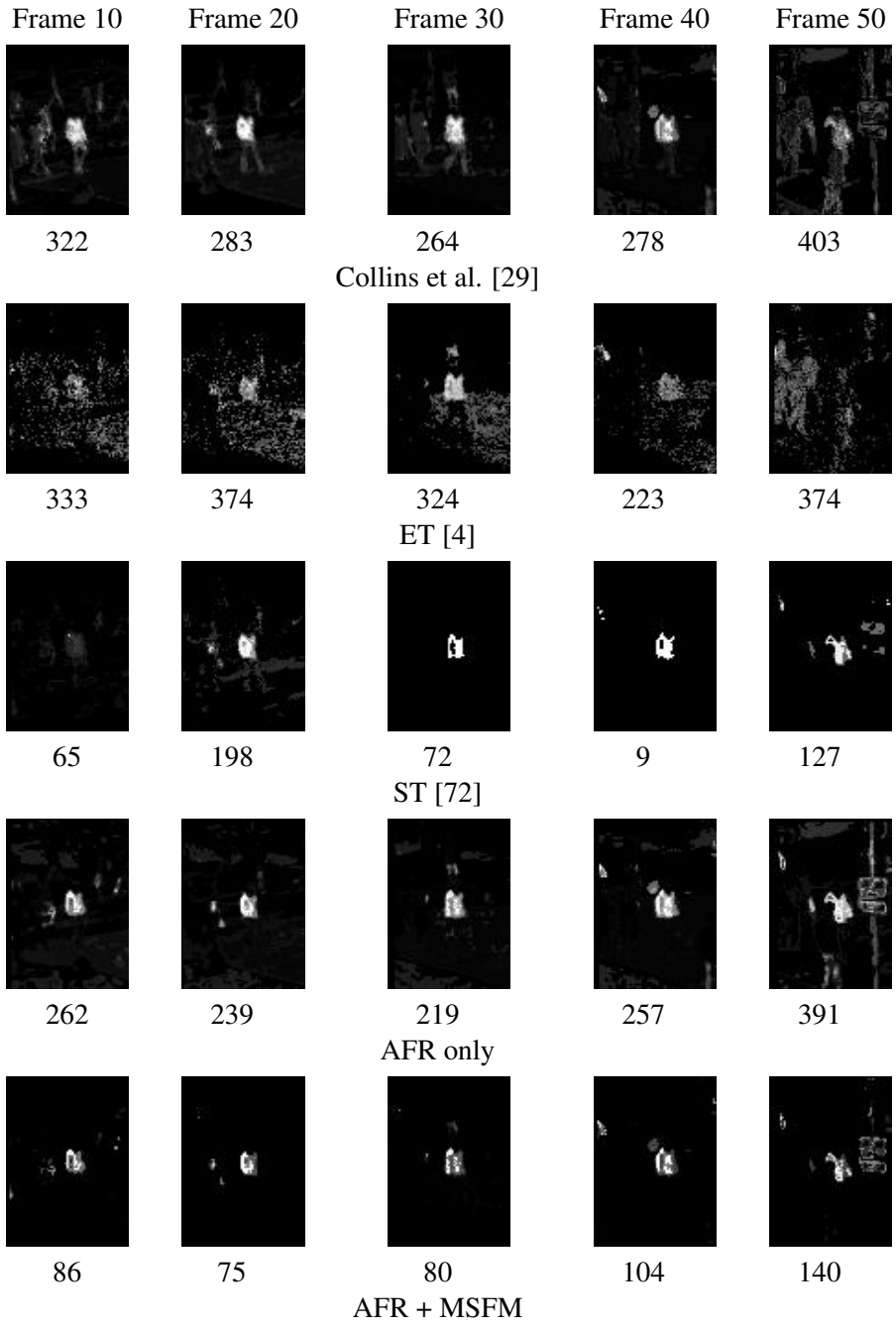
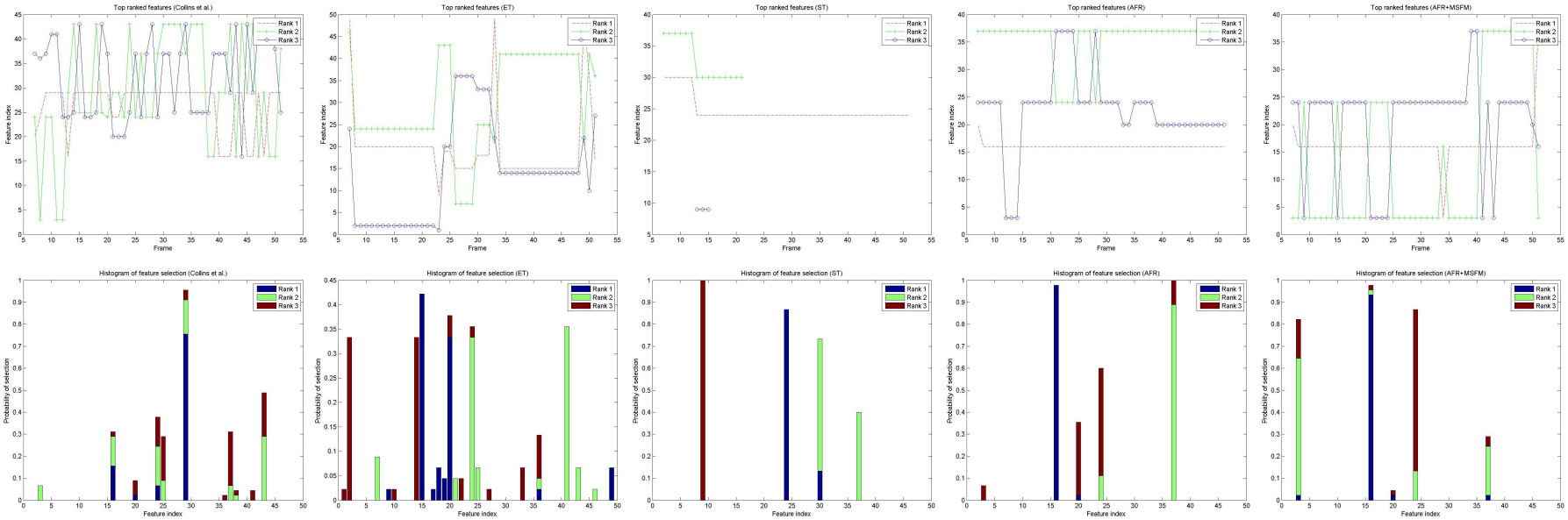


Figure 4.2: Scenario A Confidence Maps: The confidence maps correspond with the bounding box regions shown in the frames of Figure 4.1. The numbers below each image are the rounded sums of the confidences of pixels in the surround region. AFR generally shows cleaner, tighter confidence maps than both Collins et al. and Ensemble Tracking (ET) with the most salient part of the object emphasised more. The SemiBoost tracker (ST) shows sparser maps after frame 20. However, adding MSFM to AFR improves the AFR maps and further increases emphasis on the salient object region consistently for all frames, demonstrating the greater specificity of the adaptive models for AFR+MSFM.



Collins et al. ET ST AFR AFR+MSFM

Figure 4.3: Scenario A feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. Only features with non-zero weights are included. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner confidence maps (see Figure 4.2). The SemiBoost tracker (ST) showed the least variation in feature selection. From frame 20 onwards, the same single feature was effectively selected for classification, with all other features weighted zero.

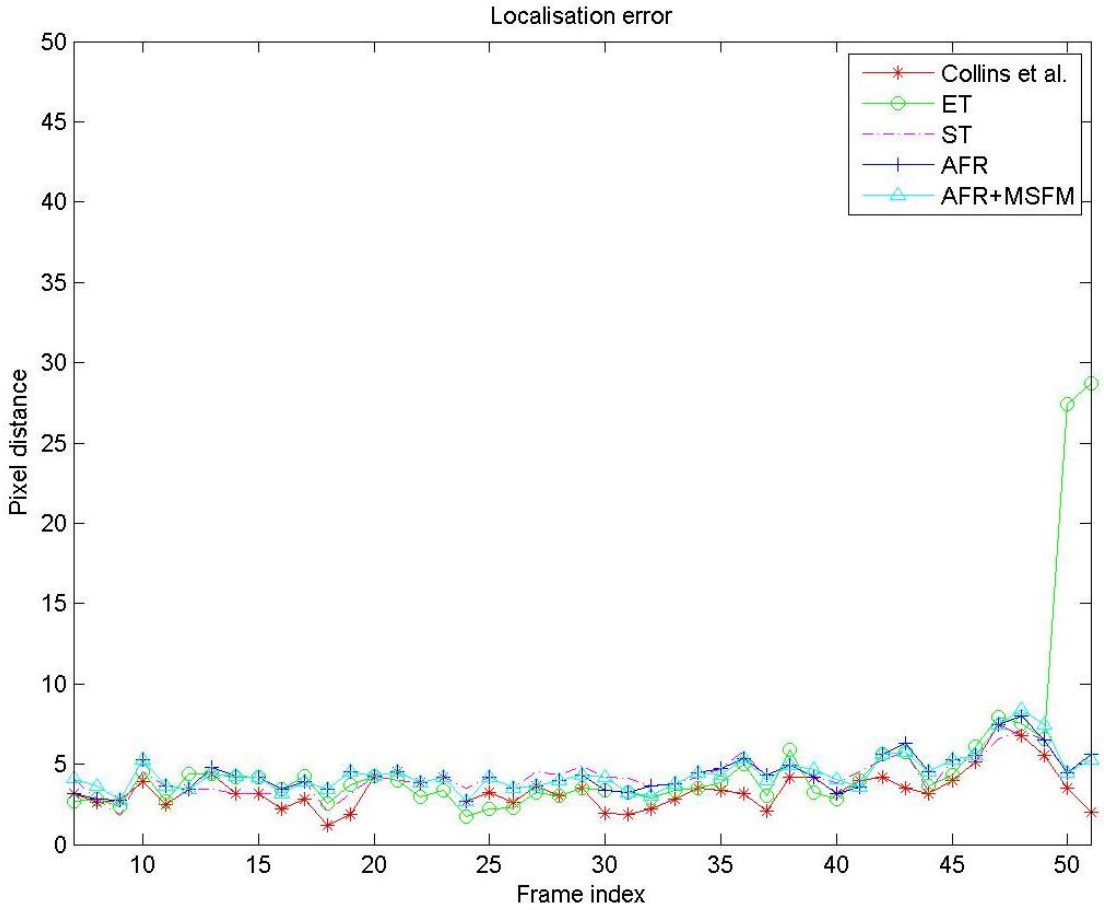


Figure 4.4: Scenario A localisation errors in pixel distance between each of the trackers and manually-labelled ground truth. All trackers performed well for this scenario, with only Ensemble Tracking deviating in the last couple of frames.

Example 1 - Brightness change

Figure 4.5 shows some key output frames from each tracker when applied to the sequence undergoing brightness change. Brightness was gradually reduced over a period of around 125 frames (around four seconds in real time) before gradually being increased at the same rate to original conditions. Specifically, Figures 4.5 (a)-(e) show tracking outputs for the trackers of Collins et al., Ensemble Tracking (ET), the SemiBoost tracker (ST), AFR and AFR+MSFM respectively. Figure 4.6 shows corresponding confidence maps generated from a weighted combination of the top three ranked features for each tracker. The SemiBoost tracker was distracted very early on, as shown in the second column. Collins et al. and Ensemble Tracking eventually failed by Frame 5867 whereas AFR and AFR+MSFM were able to maintain a lock on the target object.

The AFR tracker failed in Frame 5876, by which time AFR+MFSM was seemingly consistently displaced from the target. Interestingly however, as can be seen by a further examination in Figure 4.9 and the final column of Figure 4.5, this proved to be a precursor to tracker recovery. From just after Frame 5860, the lighting became dark enough for the tracker to no longer be able to distinguish the target; however, as can be seen in Frames 5880, 5900 and 5920 of Figure 4.9(a), the tracker appeared to attach itself to the bag being carried by the person. This was reflected by the slight offset from the center of the person as shown in the ground truth plot in Figure 4.9(b). The plot in Figure 4.9(c) shows the dominant feature for frames from 5850 onwards. It can be seen that up to the point when the target became too dark, feature number 16 was consistently the strongest feature. During this period, its corresponding feature reference model was continually updated. When the tracker lost its target, different features became dominant in turn. At this point, the reference model for feature 16 was effectively frozen since it was no longer relevant and reliable samples for update could not be collected. A different set of dominant features emerged with feature 12 particularly relevant to tracking the bag belonging to the person, up to just after Frame 5950 when the person became salient enough to be recaptured. At this point, Feature 16 was once again consistently ranked the most stable and dominant, whereupon its reference model became active again and the tracker resumed where it left off prior to its temporary failure. It is interesting to note that, due to the period of dominance of feature 12, the representation of the bag was strengthened during the darker period of the sequence via the adaptive reference models and consequently, following reacquisition of the person, the bag itself became an integral component of the representation for tracking, as exemplified by the fluctuation in highest rank between

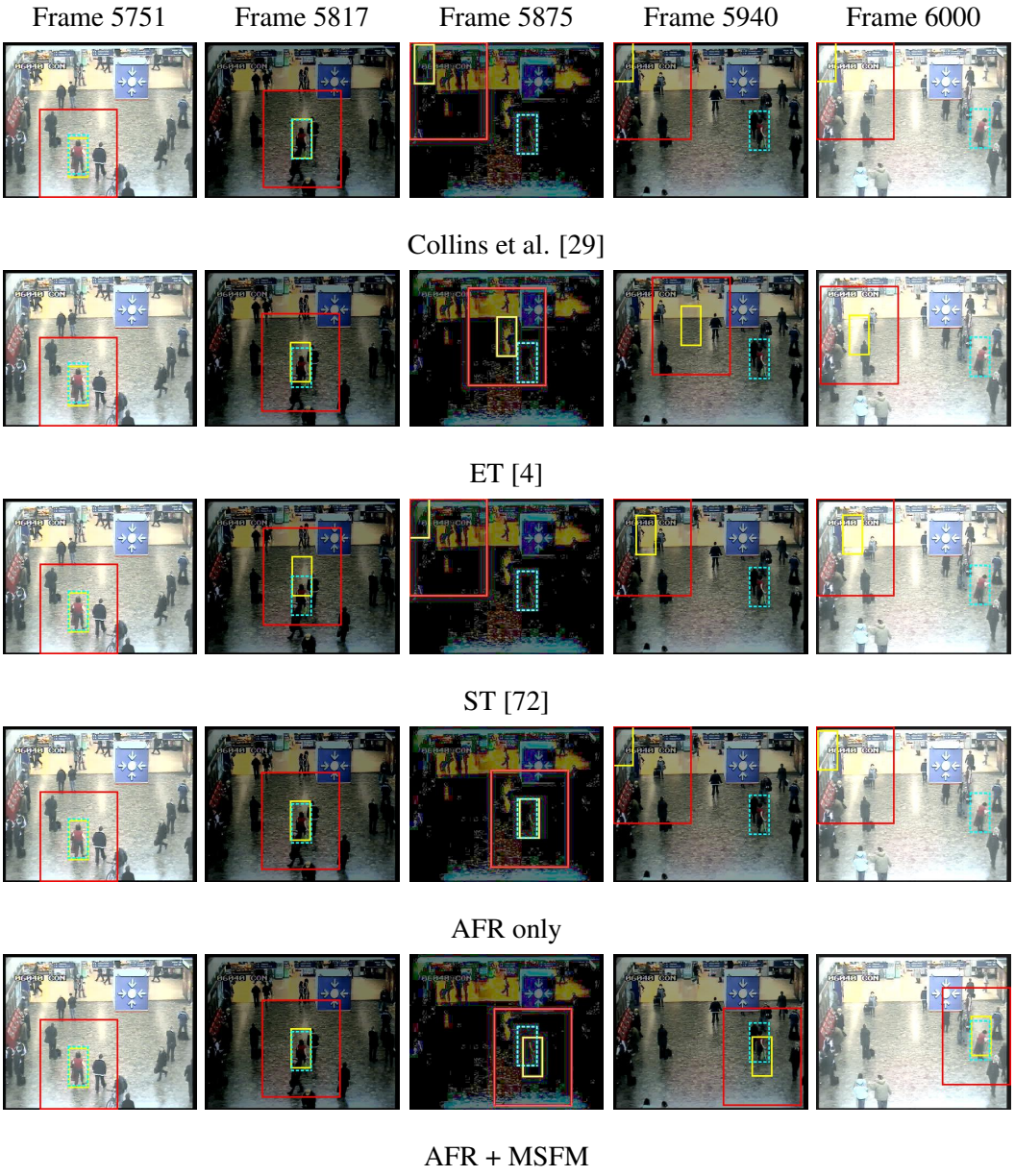


Figure 4.5: Scenario B, Example 1 Tracker Output: Severe brightness change. Frame 5875 is gamma corrected for reader clarity. Brightness was reduced by subtracting from original pixel values until around halfway through the sequence when it was increased again (from left to right respectively, images show 100%, 31%, 2%, 48% and 112% of original average pixel values). The centre-surround box shows tracker localisation for the object. The dashed blue box indicates manually labelled ground truth. Collins et al., Ensemble Tracking (ET) and the SemiBoost tracker (ST) all failed to maintain tracking. AFR managed to track for longer but failed when the scene became extremely dark. AFR+MFSM was able to adapt to track the person's bag which was still sufficiently salient, until the person became strongly visible enough to be recaptured by the tracker for the remainder of the sequence. See Figure 4.6 for corresponding confidence maps.

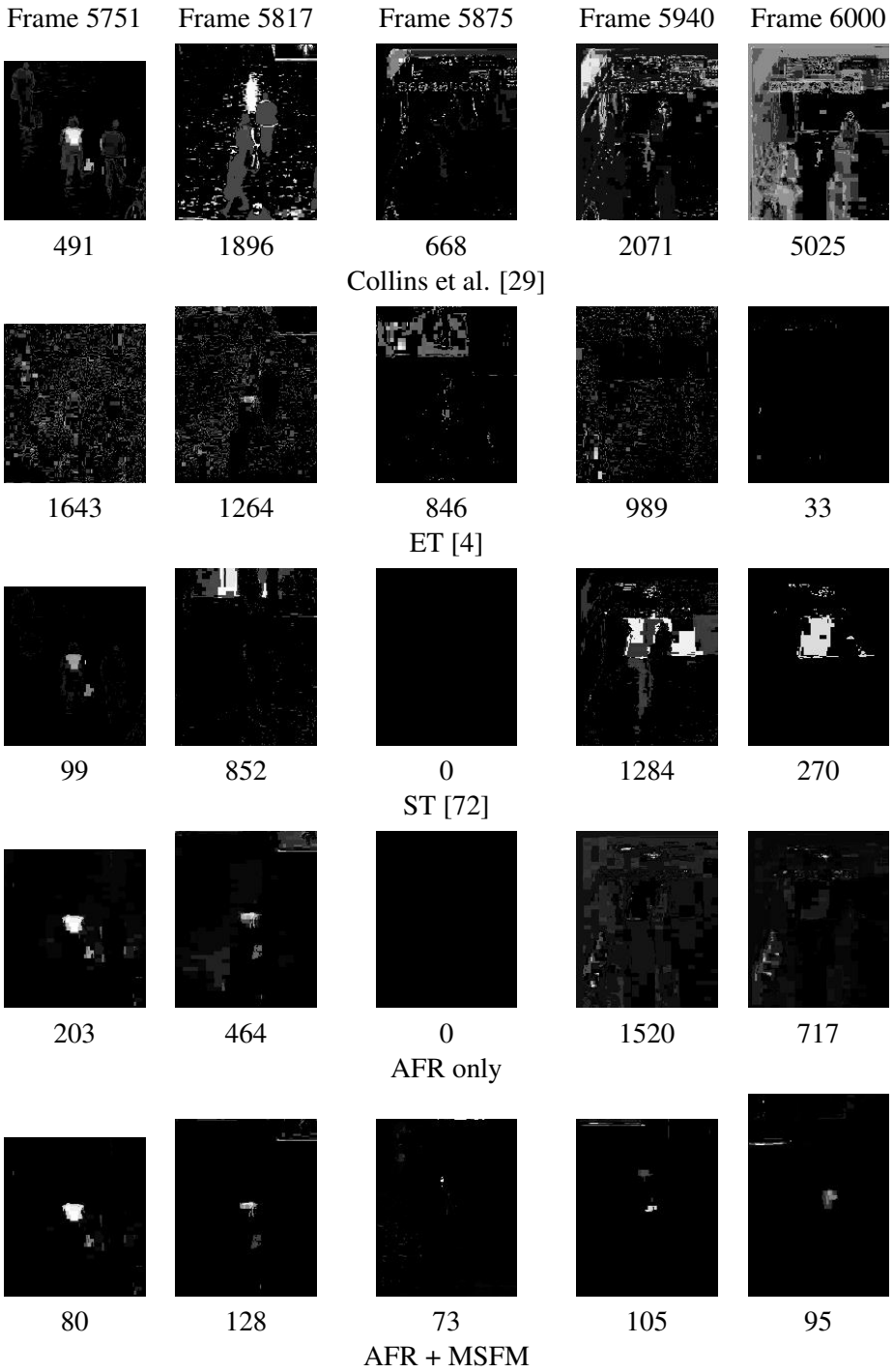
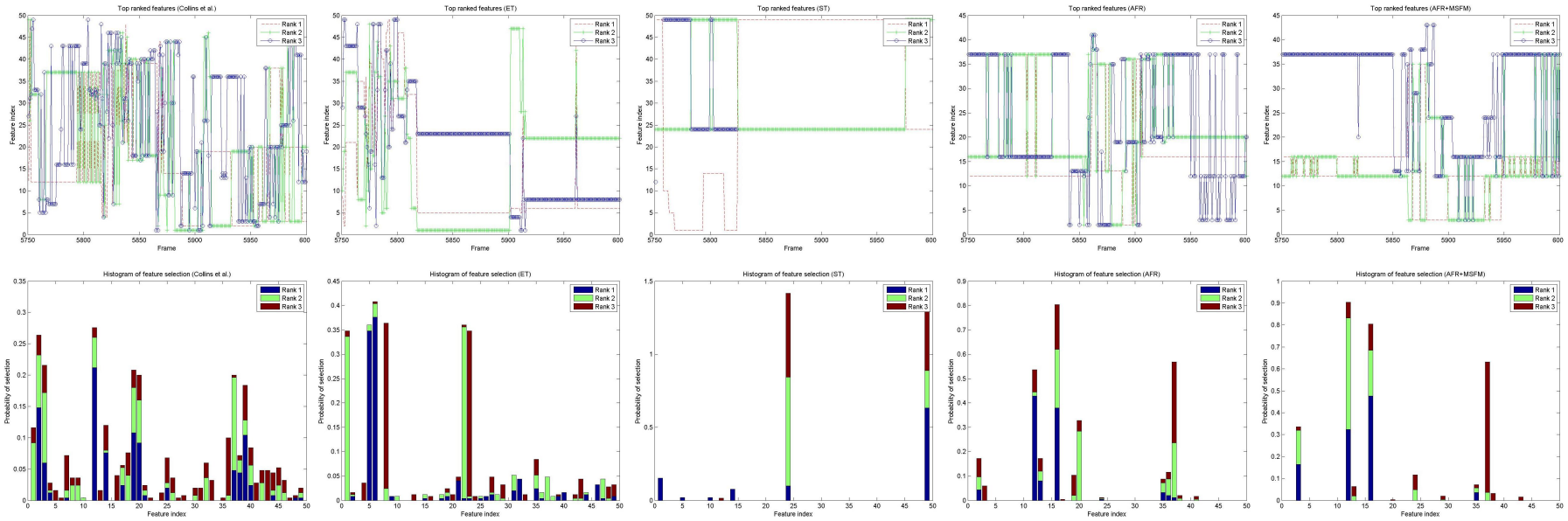


Figure 4.6: Scenario B, Example 1 Confidence Maps: Severe brightness change. The numbers below each image are the rounded sums of the confidences of pixels in the surround region. Features selected by AFR (when locked on) resulted in improved confidence maps and pixel classification over Collins et al. and Ensemble Tracking, with emphasis on the most salient object region. AFR+MSFM improved on this further still. The SemiBoost (ST) tracker showed clean maps initially but failed earliest of all.



Collins et al. ET ST AFR AFR+MSFM

Figure 4.7: Scenario B, Example 1 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner confidence maps (see Figure 4.6). The SemiBoost tracker (ST) showed least variation in feature ranking, but the tracker failed the earliest.

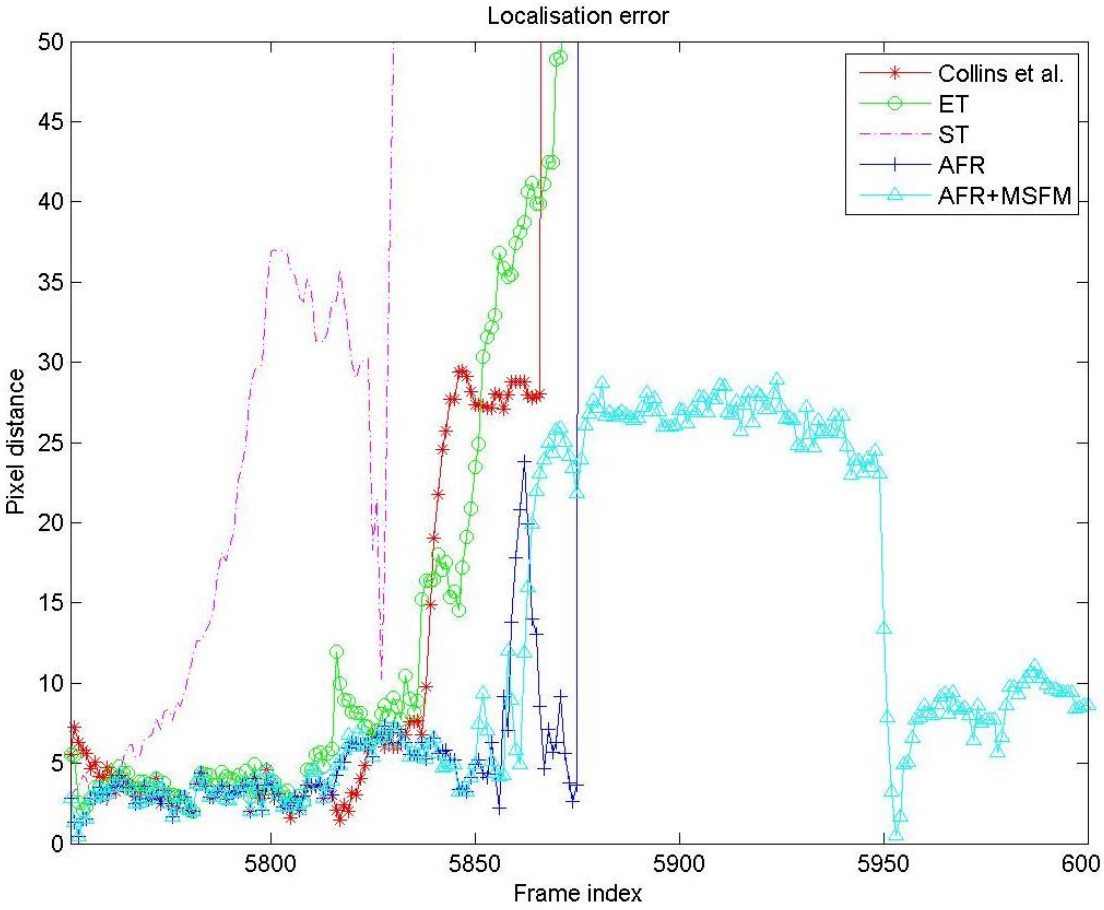


Figure 4.8: Scenario B, Example 1 localisation errors in pixel distance between each tracker and manually-labelled ground truth. The SemiBoost tracker (ST) failed near the beginning of the sequence whereas Collins et al. and Ensemble Tracking (ET) were unable to cope with the darkest portion of the sequence and lost track. AFR failed shortly afterwards. However, AFR+MFSM was able to latch onto a still-salient part of the target object (a carrier bag) until sufficient brightness was restored for the tracker to reacquire the target object around frame 5950.

features 16 and 12 (for person and bag respectively) from frame 5950 onwards. This illustrates the advantages of adaptive reference models in maintaining the association between object and features in a dynamic way under changing conditions.

As can be seen from the confidence maps in Figure 4.6, the most salient part of the target object (the red coat) was strongly reflected since AFR ranked the more appropriate features more highly (while the tracker was still attached to the target). Using MSFM in addition further improved the confidence maps since the feature reference models were updated to reflect the more strongly discriminative feature values, which in turn contributed more to greater classification accuracy.

Figure 4.7 shows the feature selection statistics for the top three ranked features for this example. Here, compared to Collins et al. and Ensemble Tracking (ET), we see a less haphazard ranking of features when using AFR or AFR+MSFM. The methods resulted in tighter, more representative confidence maps and more robust tracking as seen in Figure 4.5 (d) and (e). The SemiBoost tracker (ST) had the most rigid record of feature ranking; however, early failure of the tracker renders this moot and suggests that this seemingly less haphazard ranking behaviour is not related to the selection of the most appropriate features in practice. Figure 4.8 gives an overview of tracking performance by plotting tracker accuracy as pixel displacements from manually-labelled ground truth.

Example 2 - Colour change

Figure 4.10 shows some output frames from each tracker when applied to the sequence undergoing colour change. The red channel was gradually reduced and the blue channel increased over a period of around 125 frames (around four seconds in real time). Specifically, Figures 4.10 (a)-(e) show tracking outputs for the trackers of Collins et al., Ensemble Tracking (ET), the SemiBoost tracker (ST), AFR and AFR+MSFM respectively. Figure 4.11 shows the corresponding confidence maps. Ensemble Tracking and the SemiBoost tracker were distracted prior to halfway through the sequence whereas Collins et al. became attached to another moving person later on. Both AFR and AFR+MSFM were able to maintain a lock on the target object for the duration of the sequence. This is further illustrated in the ground truth plot in Figure 4.13. As can be seen in Figure 4.11, AFR and AFR+MFSM consistently showed the least noisy confidence maps, with the adaptive reference models of AFR+MFSM helping to further emphasise the most salient discriminators of the target object.

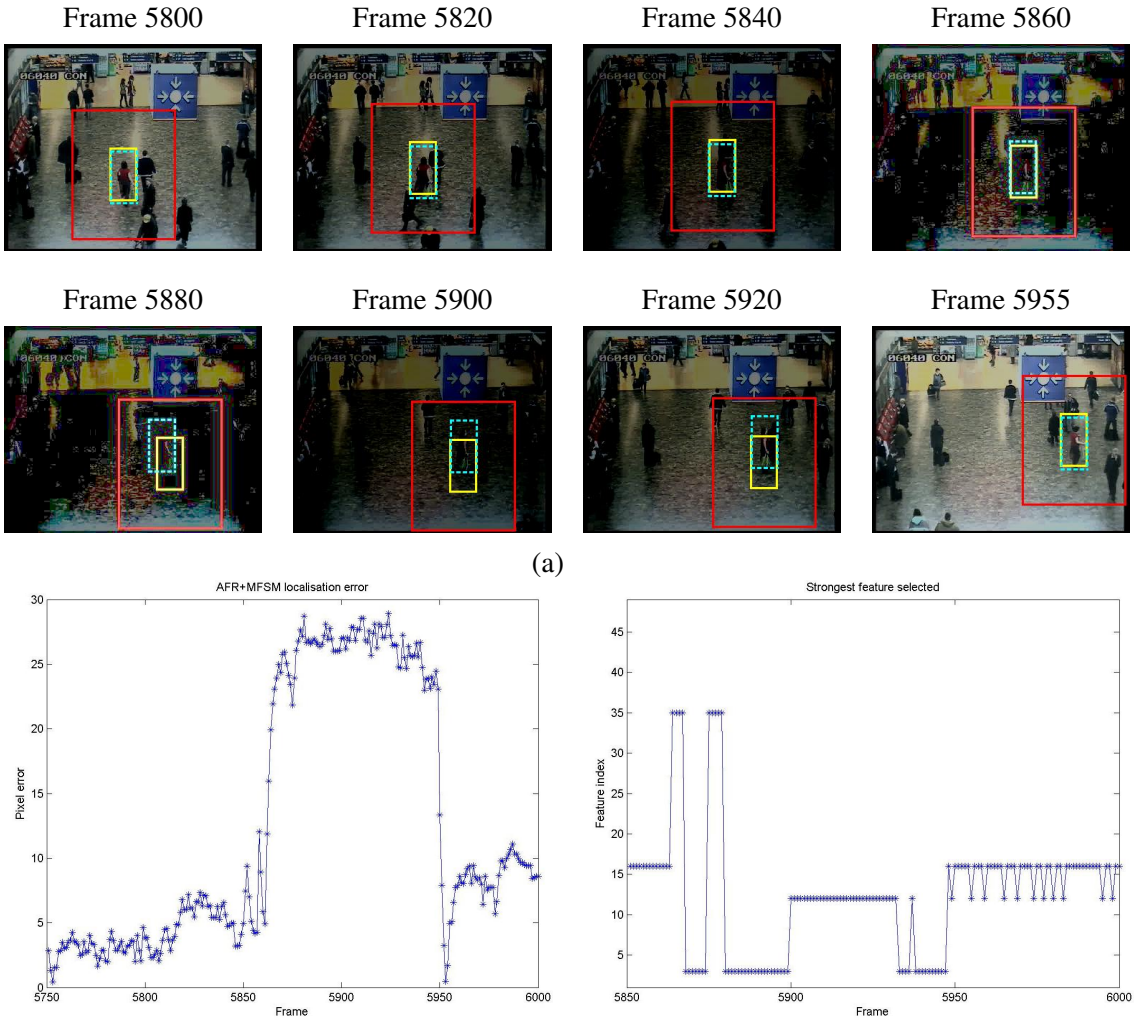


Figure 4.9: Recovery of AFR+MSFM after distraction. (a) Selected frames with ground-truth (dashed blue box) and tracker output (yellow box) superimposed. (b) Error in tracking position relative to ground-truth. (c) Most dominant feature from frame 5850 onwards. Shortly after frame 5860 the person lost saliency due to severe low lighting and the dominant feature changed from 16 to switching between 35 and 3, as can be seen from Plot (c). However, from around frame 5880, the tracker was able to attach itself to a still-salient bag being carried by the person, which was well characterised by feature 12 at that brightness range. As scene brightness returned, feature 3 reasserted itself around frame 5930. Around Frame 5950 the person became visible enough for the tracker to quickly reattach itself via the previously dominant feature 16, whose reference model asserted itself to reacquire the target object. From this point the bag was also consistently emphasised via feature 12, since via MSFM its representation was strengthened during the darker period, with the result that both relevant features (16 and 12) reinforced each other in characterising the tracked person.

Figure 4.12 shows the statistics for the top three ranked features for this example. Again, here we see less haphazard feature ranking from AFR than Collins et al. and Ensemble Tracking (ET), with AFR+MFSM improving on this still further. Again, the SemiBoost tracker (ST) showed the least variation, but this did not translate to accuracy of tracking in this experiment. This again suggests that the feature ranking behaviour was not as well correlated with the use of the most appropriate features, unlike for AFR and AFR+MFSM.

4.3.5 Scenario C: Tracking under occlusion

In this scenario, we track a moving person at a distance from the camera in a sunny outdoor environment as they undergo severe occlusion by other moving people. This allows us to observe the ability of each tracker to deal both with temporary severe occlusion as well as sunny outdoor environments exhibiting strong shadowing effects which can lead to sudden, transient changes in appearance.

Figure 4.14 shows some key output frames from each tracker when applied to this sequence. Figure 4.15 shows the corresponding confidence maps. Specifically, Figures 4.14 (a)-(e) show tracking output frames and confidence maps for the top ranked feature generated by Collins et al., Ensemble Tracking (ET), the SemiBoost tracker (ST), AFR and AFR+MSFM respectively. As can be verified by the ground truth plot in Figure 4.17, Collins et al., Ensemble Tracking and the SemiBoost tracker all failed within a few frames, whereas AFR and AFR+MSFM were able to continue tracking. Frame 49 shows the target object undergoing significant occlusion by another moving object. Shortly after, the AFR-only tracker began to get attracted to the distractor. However, AFR+MSFM remained locked onto the target object due to the up-to-date adapted feature models which were able to counteract the tendency for the frame-specific model to drift. As can be seen in Figure 4.15, as for previous experiments the AFR maps were cleaner and more representative of the target when locked on, with AFR+MSFM further emphasising the most salient discriminators of target appearance and helping to maintain tracking even when AFR had failed.

The feature ranking statistics are shown in Figure 4.16. Although the Collins et al. statistics showed a less haphazard pattern than usual, this can be attributed to early failure and attachment to a relatively static region of the scene. The SemiBoost tracker selected the same single feature for the entire sequence due to its failure at the beginning of the sequence, latching onto a static portion of the background. AFR and AFR+MSFM again demonstrated a better relationship

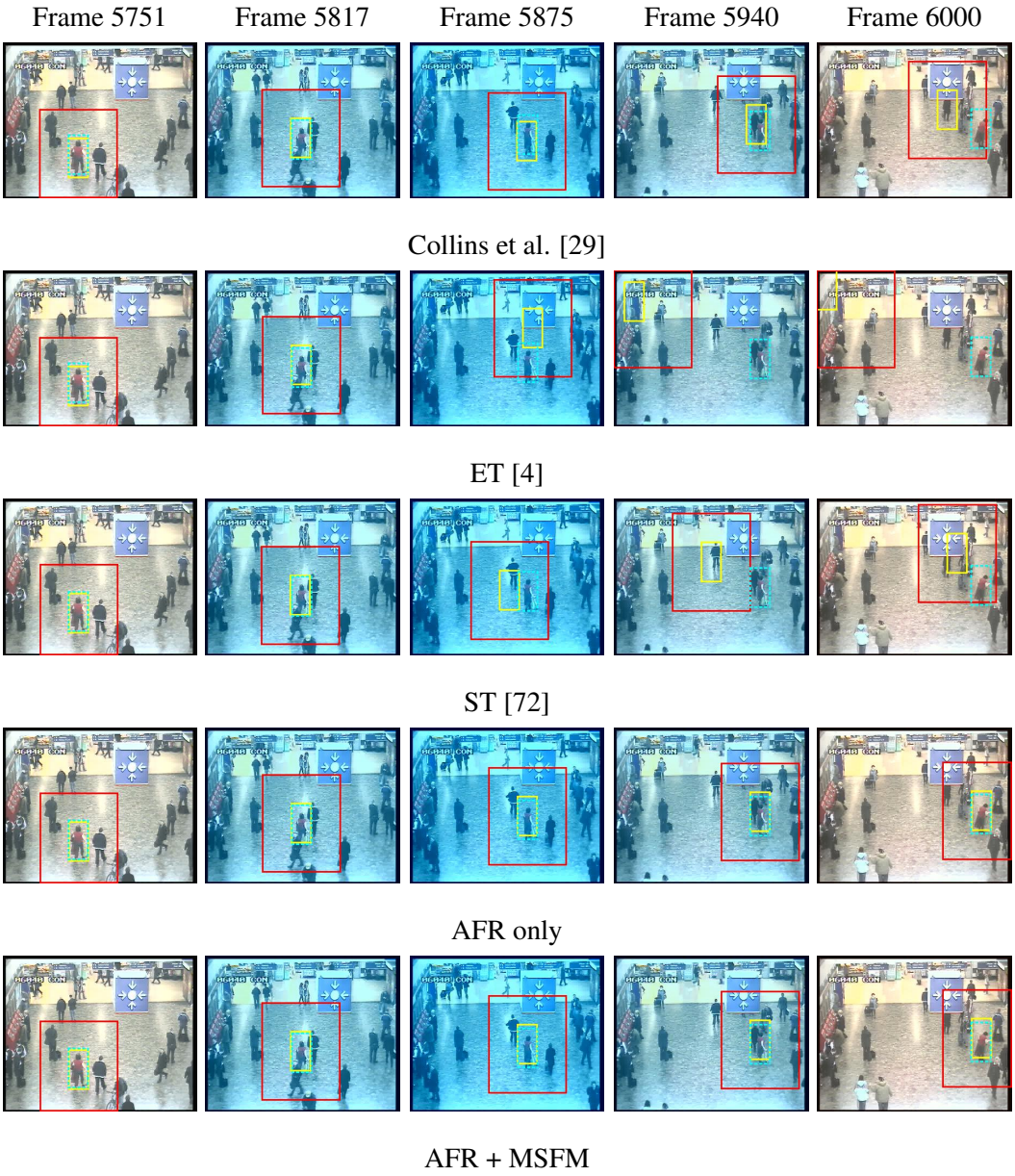


Figure 4.10: Scenario B, Example 2 Tracker Output: Severe illumination change. The red and blue pixel values were modified (from left to right respectively) to 100%, 64%, 39%, 74% and 107% (for red) and 100%, 118%, 128%, 115% and 98% (for blue) of the original average pixel values. The centre-surround box shows tracker localisation for the object. The dashed blue box indicates manually labelled ground truth. Ensemble Tracking (ET) and the SemiBoost tracker (ST) both failed around the frame 5850 mark, with Collins et al. becoming distracted around 100 frames later. AFR and AFR+MFSM both maintained track for the duration of the sequence. See Figure 4.11 for corresponding confidence maps.

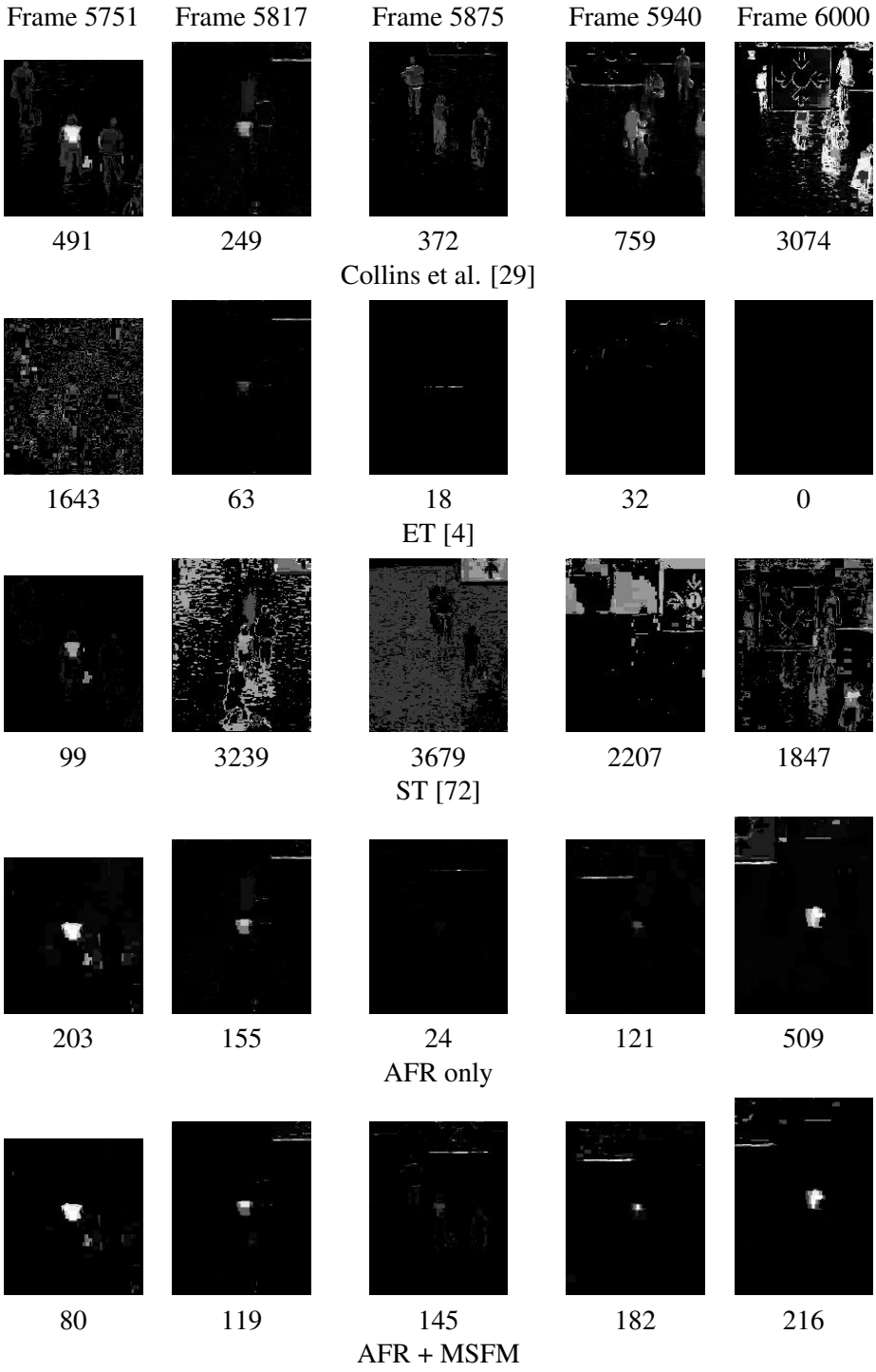
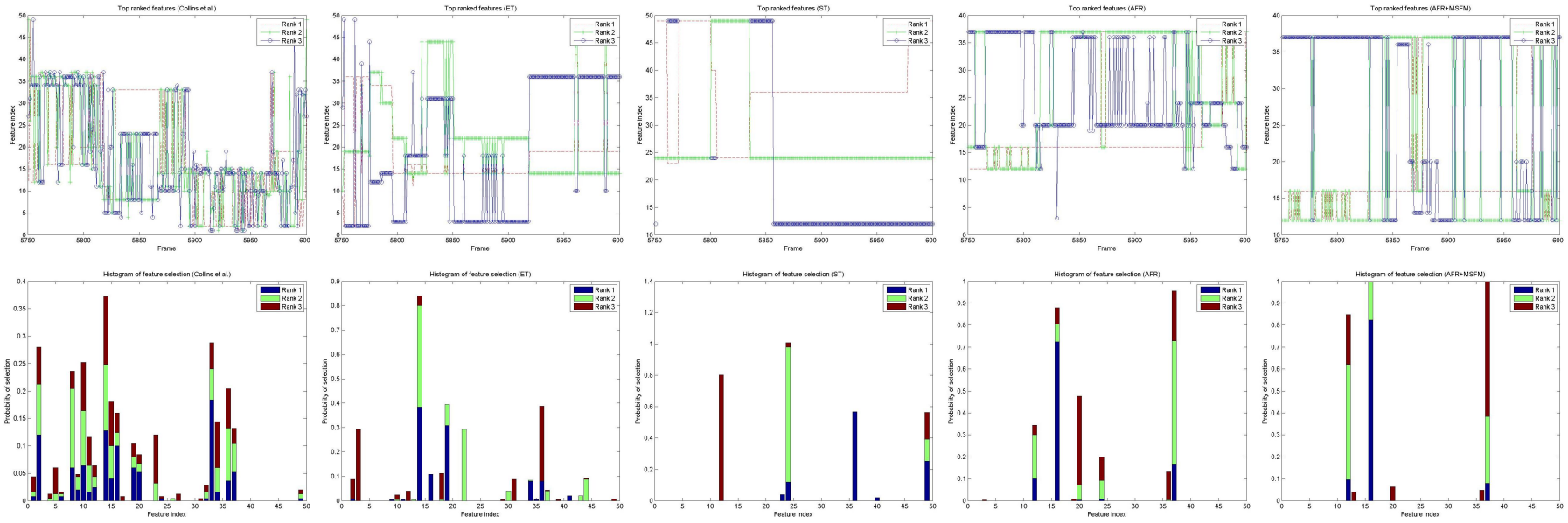


Figure 4.11: Scenario B, Example 2 Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. While the SemiBoost tracker showed a clean confidence map in the first frame before failing, the AFR and AFR+MFSM trackers showed the most useful confidence maps consistently throughout the sequence with the target object more strongly emphasised. The adaptive reference models of AFR+MFSM helped to further strengthen the most salient parts of the target object.



Collins et al. ET ST AFR AFR+MSFM

Figure 4.12: Scenario B, Example 2 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner, more relevant confidence maps (see Figure 4.11). The SemiBoost tracker (ST) showed a similar rigidity as for previous experiments, but this did not translate to tracking accuracy in practice.

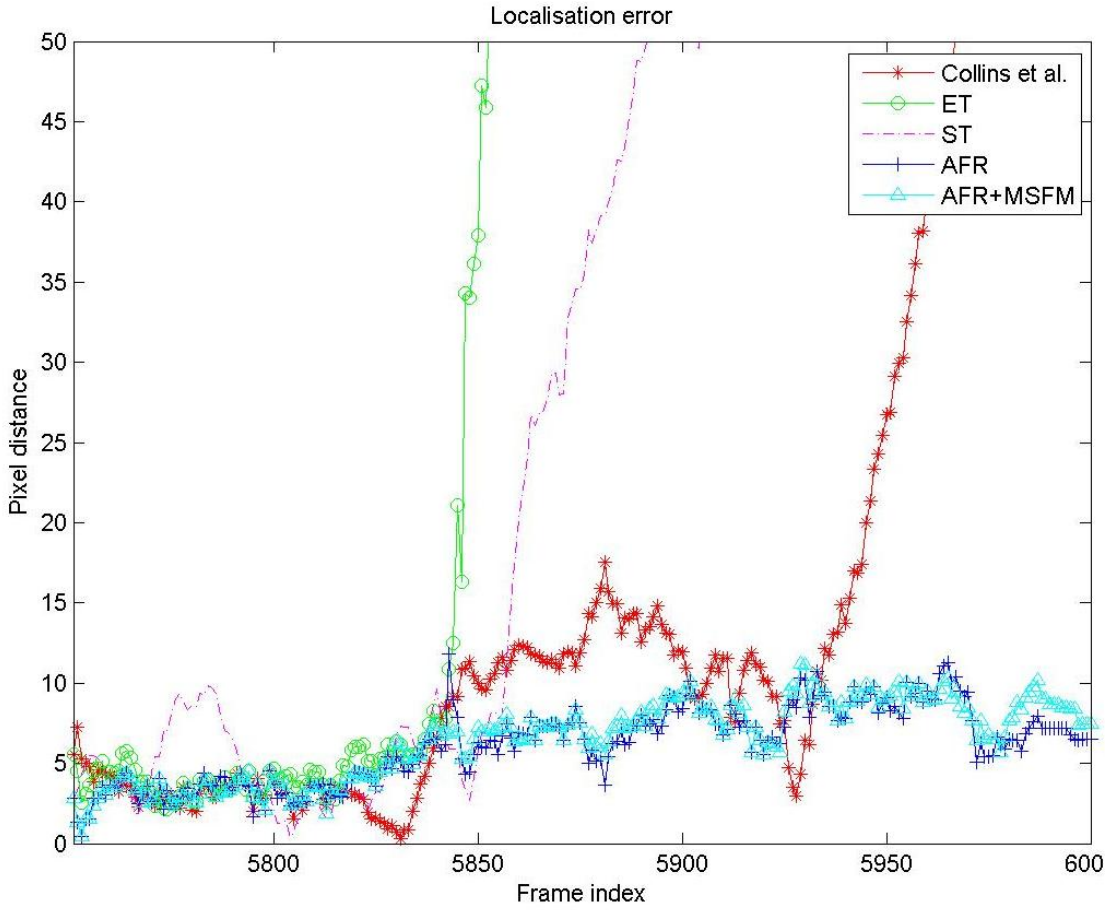


Figure 4.13: Scenario B, Example 2 localisation errors in pixel distance between each tracker and manually-labelled ground truth. Ensemble Tracking (ET) and the SemiBoost tracker (ST) both failed before the halfway point of the sequence, whereas Collins et al. failed lost track around frame 5925. Both AFR and AFR+MFSM were able to maintain tracking for the duration of the sequence.

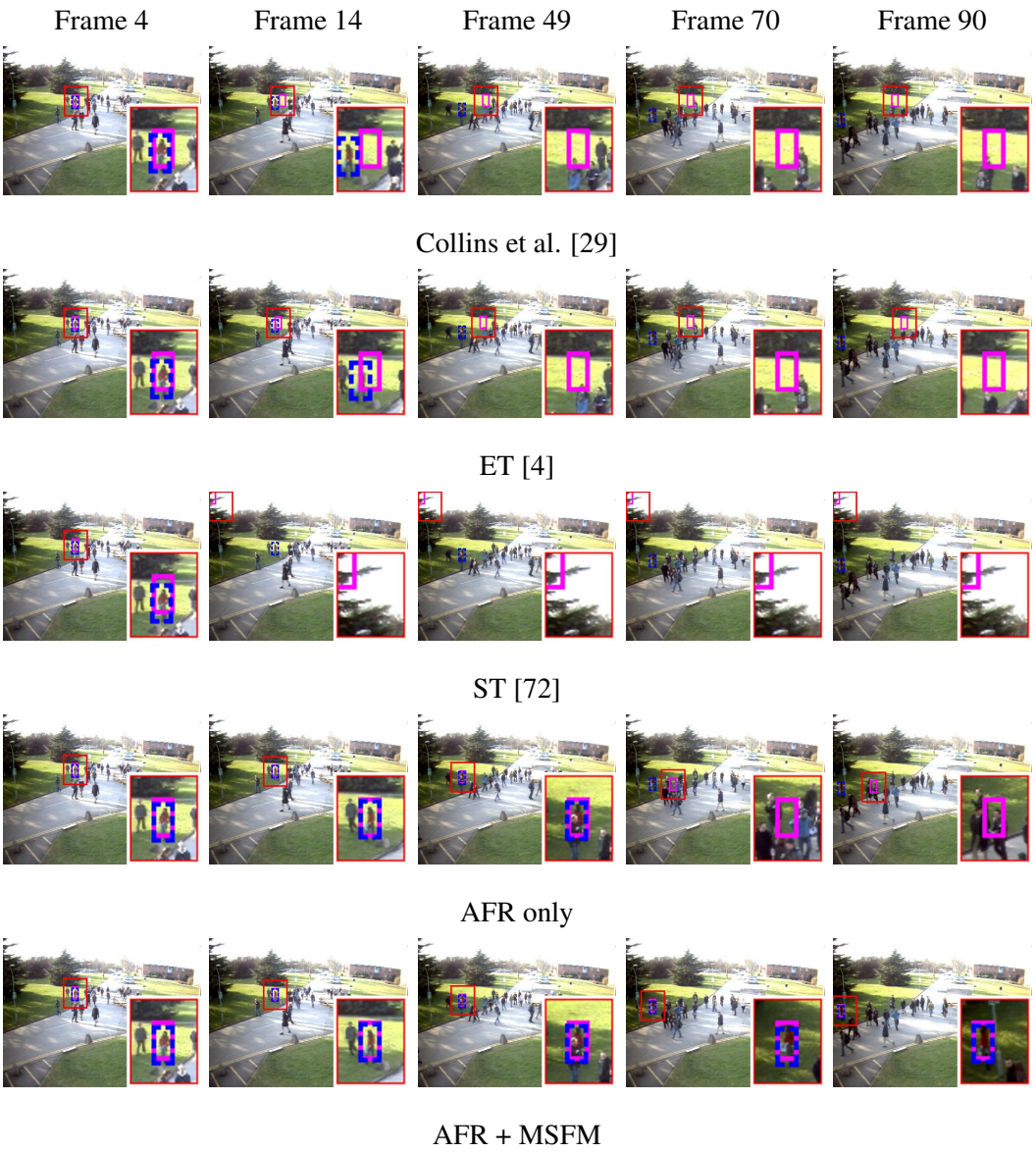


Figure 4.14: Scenario C Tracker Output: Tracking under occlusion. The dashed dark blue box shows ground truth. Collins et al., Ensemble Tracking (ET) and the SemiBoost tracker (ST) failed within a few frames, whereas AFR and AFR+MSFM continued to track successfully. AFR began to fail around Frame 52 due to occlusion (see Figure 4.17) but AFR+MSFM resisted distraction due to up-to-date feature reference models.

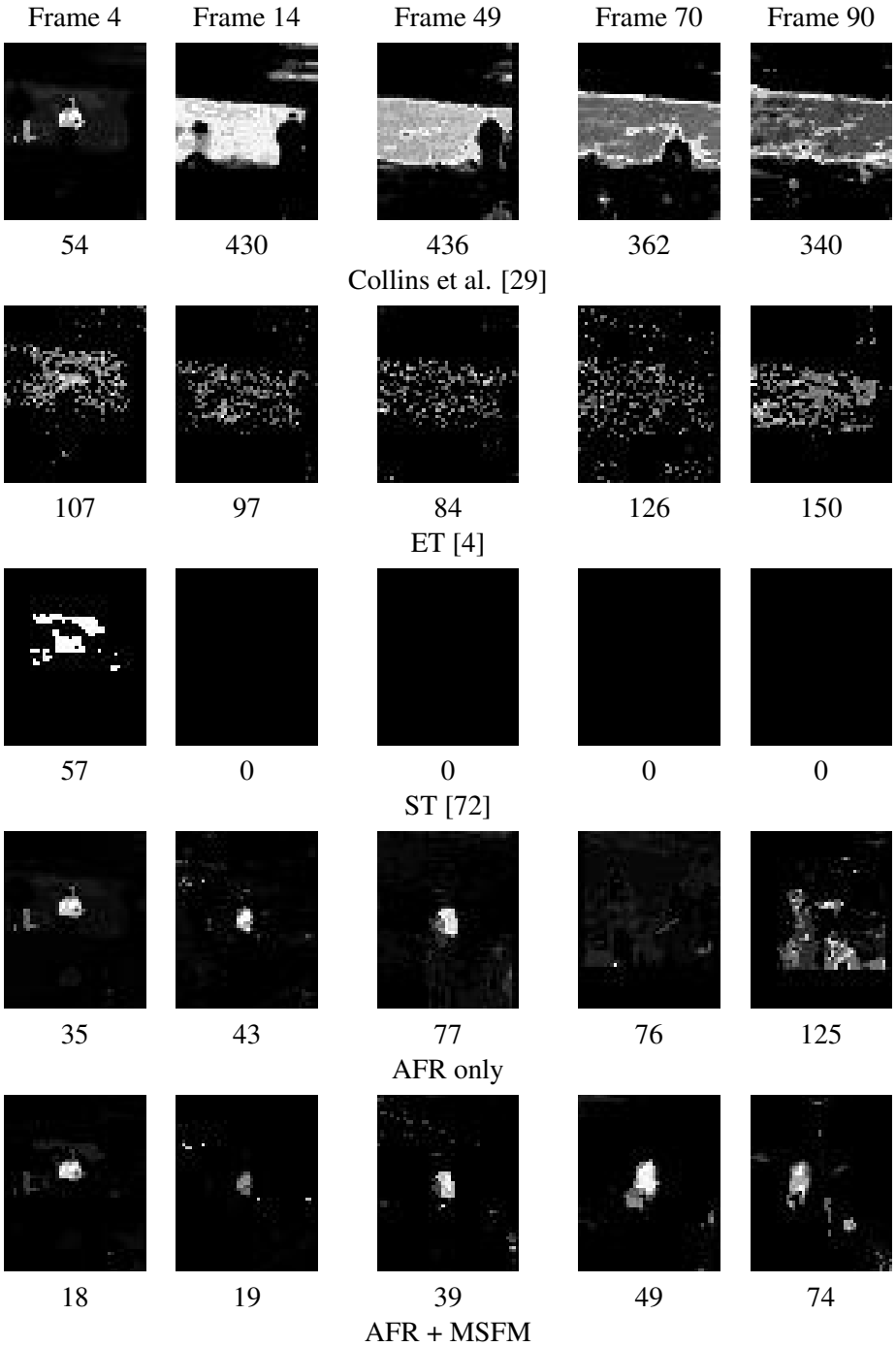


Figure 4.15: Scenario C Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. The top three rows represent trackers that failed early and so expectedly depict noisy or empty maps bearing no relevance to the tracking target. AFR maps were clean and representative up until failure, whereas AFR+MFSM again improved on AFR by further reducing noise and consequently improving actual tracking performance.

between ranked features and actual performance, with the latter providing meaningful support to the former in improving actual performance.

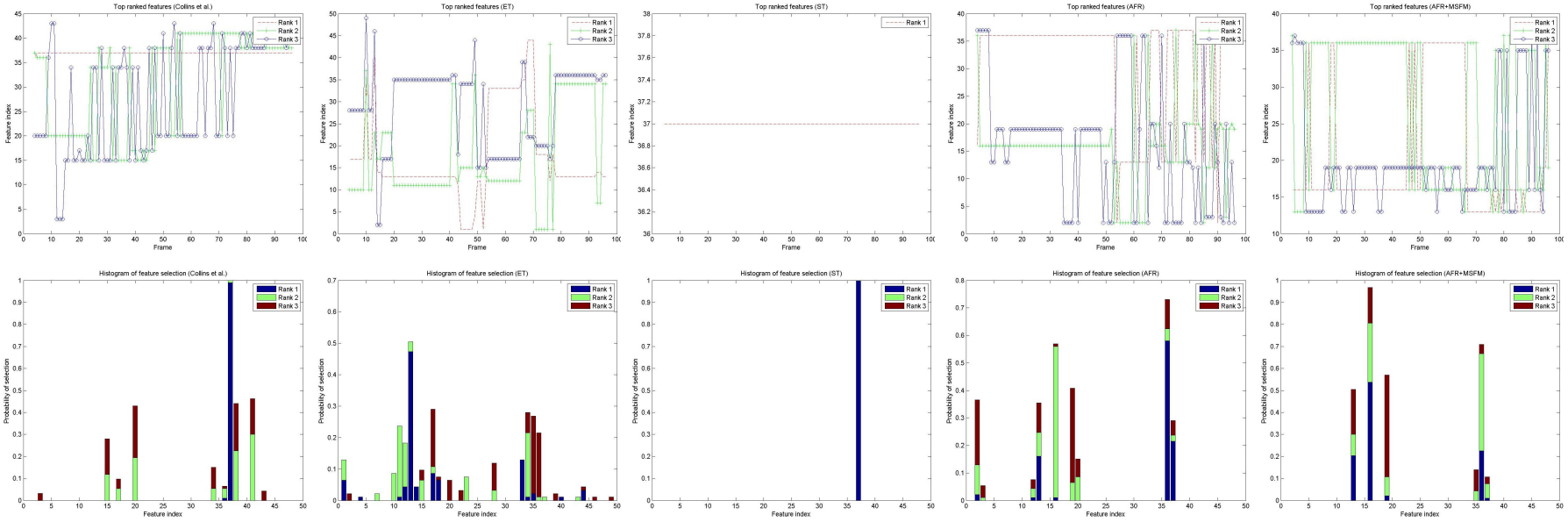
Figure 4.16 shows the feature selection statistics of the top three ranked features for this outdoor scene. Again, Collins et al. and Ensemble Tracking (ET) resulted in less stable and more haphazard feature selection patterns. AFR and AFR+MSFM continued to show greater stability in feature selection. Using AFR only, the tracker was distracted after the target object underwent occlusion but the up-to-date feature models of AFR+MSFM maintained feature selection stability and tracking accuracy. From around Frame 52 onwards the AFR-only tracker was thrown by the occlusion as a result of pixels from the distractor polluting foreground samples and causing fluctuations in feature selection and eventual tracking failure. However, the benefit of MSFM in maintaining a relevant reference of target object appearance and offsetting the contribution of undesired pixels in ranking features was evident by the continued stability and relevance of the features selected during occlusion.

4.3.6 Scenario D: Tracking under severe occlusion and illumination changes

In this scenario we track a moving person at a distance from the camera in a crowded scene whilst undergoing both significant occlusion and lighting changes. Similarly to Scenario B (Section 4.3.4), we have two examples of changing illuminant for this scenario; the first showing a change in brightness and the second a change in colour. These changes are simulated by gradually modifying the original pixel values in sequence frames over time. Again, for brightness changes, these modifications are then gradually reversed to the original conditions.

Example 1 - Brightness change

Figure 4.18 shows some example output frames from each tracker when applied to the sequence undergoing brightness change. Brightness was gradually reduced over a period of around 100 frames before gradually being increased at the same rate to original conditions. Specifically, Figures 4.18 (a)-(e) show tracking outputs for Collins et al., Ensemble Tracking (ET), the Semi-Boost tracker (ST), AFR and AFR+MSFM respectively. Figure 4.19 shows the corresponding confidence maps. As can be verified from the ground truth plot in Figure 4.21, Collins et al. and Ensemble Tracking became distracted around the same time, shortly after frame 90 when the target object was both heavily occluded and lacking in saliency due to a significant lack of brightness. AFR and AFR+MFSM were both able to successfully hold the target for the duration



Collins et al. ET ST AFR AFR+MSFM

Figure 4.16: Scenario C feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than Ensemble Tracking (ET), with the AFR+MFSM again further improving on AFR alone. This resulted in cleaner confidence maps (see Figure 4.15). Collins et al. showed less fluctuation than normal, but this is attributable to its attachment to a relatively static portion of the background following tracking failure. The SemiBoost tracker (ST) selected the same single feature for the whole sequence, but again this was largely due to tracking failure occurring at the beginning of the sequence.

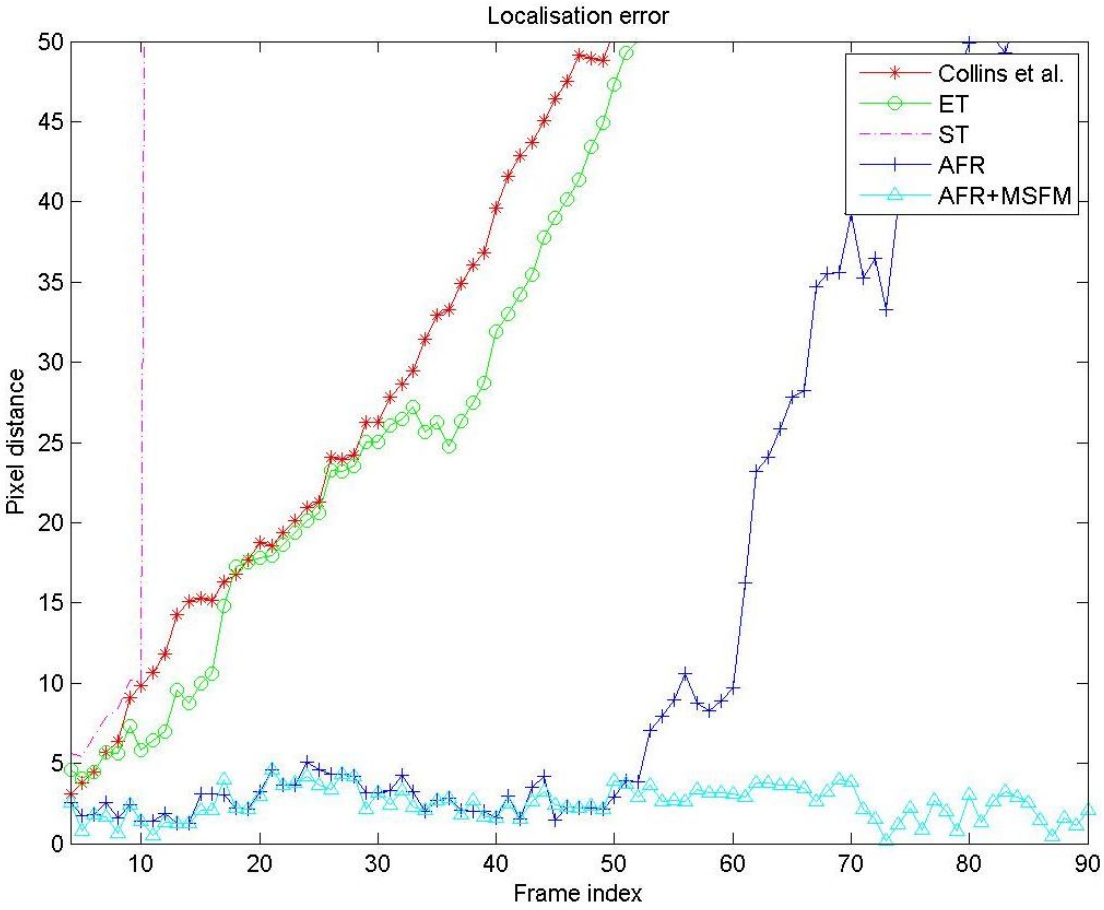


Figure 4.17: Scenario C localisation errors in pixel distance between each tracker and manually-labelled ground truth. Collins et al., Ensemble Tracking (ET) and the SemiBoost tracker (ST) failed almost instantly, whereas AFR was able to track the target before being distracted by a significant moving occluder shortly after frame 50. AFR+MFSM maintained a lock on the target for the duration of the sequence.

of the sequence. Their corresponding confidence maps in Figure 4.19 again show the familiar relationship of relatively clean maps from AFR with the addition of MSFM providing further improvement by way of emphasising the most salient feature values. The supportive nature of MSFM for AFR was also reflected in the feature selection statistics of Figure 4.20 where the more controlled pattern of feature selection correlated with the quality of the resulting confidence maps and actual tracker performance.

In this experiment, the SemiBoost tracker was also able to successfully track the target object for the duration and as before showed significant rigidity in feature ranking and selection (with the same feature being ranked highest throughout). The resulting confidence maps were the cleanest of all. However, given previous results, this is more likely to be through fortune rather than any structural relationship between the feature ranking/selection approach and the most relevant features for the sequence in particular.

Example 2 - Colour change

Figure 4.22 shows some key output frames from each tracker when applied to the sequence undergoing colour change. The red channel was gradually reduced and the blue channel gradually increased over a period of around 100 frames. Specifically, Figures 4.22 (a)-(e) show tracking outputs for Collins et al., Ensemble Tracking (ET), the SemiBoost tracker (ST), AFR and AFR+MSFM respectively. Figure 4.23 shows the corresponding confidence maps.

As can be verified from the ground truth plot in Figure 4.25, this time the SemiBoost tracker failed shortly after frame 10, losing the target completely. The other four trackers continued up to around frame 95, where the Avidan tracker suddenly lost the target object. The Collins et al. and AFR trackers also became attracted to a nearby distractor partially occluding the target object; however, the AFR tracker was able to recover within around 15 frames. The Collins et al. tracker also recovered around frame 120. This was likely helped by the fact that the target object stopped moving, although surrounding distractors continued to move and cast shadows on the target object and the lighting continued to change. AFR+MFSM successfully maintained a lock on the object for the duration of the sequence.

The confidence maps in Figure 4.23 were relatively clean for all trackers when they were locked on, with failed frames showing noisy confidence maps indicative of the models' poor correlation with the image data given the bounding box position. The AFR+MFSM combination again provided the balance to maintain tracking via a more appropriate selective adaptation of

Figure 4.18: Scenario D, Example 1 Tracker Output: Tracking under occlusion and lighting change. Frames 80 and 120 were gamma corrected for reader clarity. Brightness was reduced by subtracting from original pixel values until around halfway through the sequence when it was increased again (from left to right respectively, images show 100%, 48%, 18%, 20% and 51% of original average pixel values). The dashed dark blue box shows ground truth. Collins et al. and Ensemble Tracking (ET) failed around frame 90, whereas the SemiBoost tracker (ST), AFR and AFR+MSFM all continued to maintain a lock successfully for the duration of the sequence.

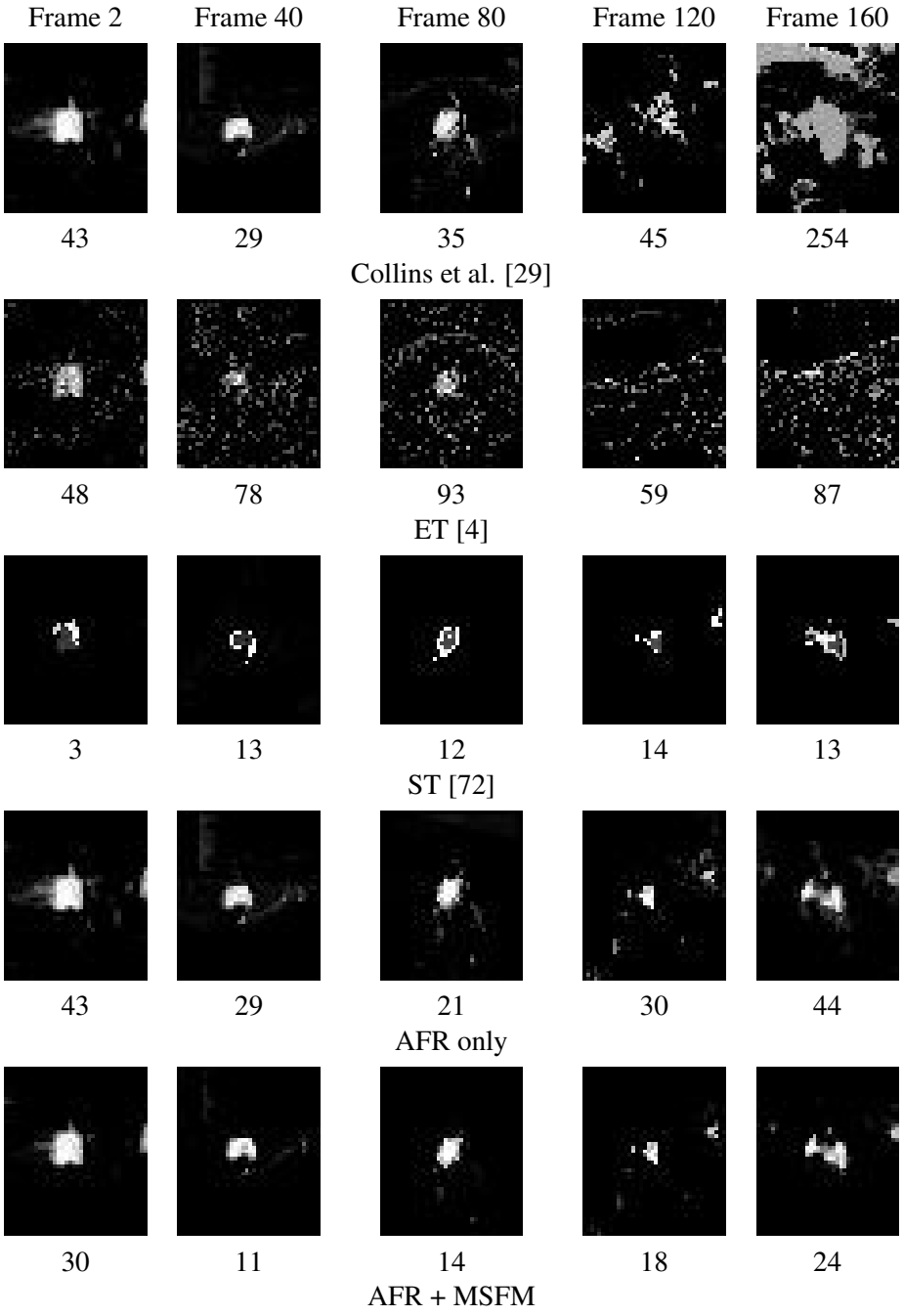
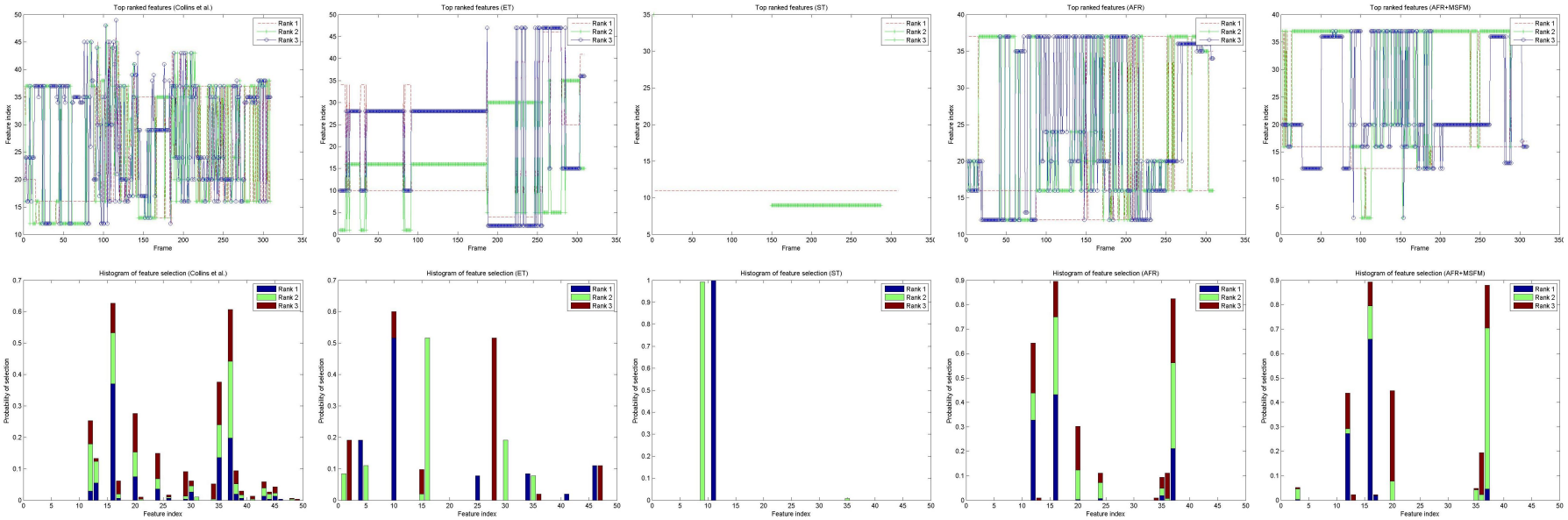


Figure 4.19: Scenario D, Example 1 Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. The SemiBoost tracker (ST), AFR and AFR+MFSM all showed the most consistently representative confidence maps, with MFSM again showing its ability so support AFR in improving pixel classification. The SemiBoost tracker showed the cleanest maps as a result of its lack of flexibility in feature ranking (see Figure 4.20).



Collins et al. ET ST AFR AFR+MSFM

Figure 4.20: Scenario D, Example 1 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. The features selected for AFR and AFR+MSFM were better discriminators and prone to fewer fluctuations than both Collins et al. and Ensemble Tracking (ET). This resulted in cleaner confidence maps (see Figure 4.19), with AFR+MFSM again demonstrating meaningful improvement over AFR alone. The SemiBoost tracker (ST) again showed significant rigidity in feature ranking, which in this example proved adequate for tracking to succeed.

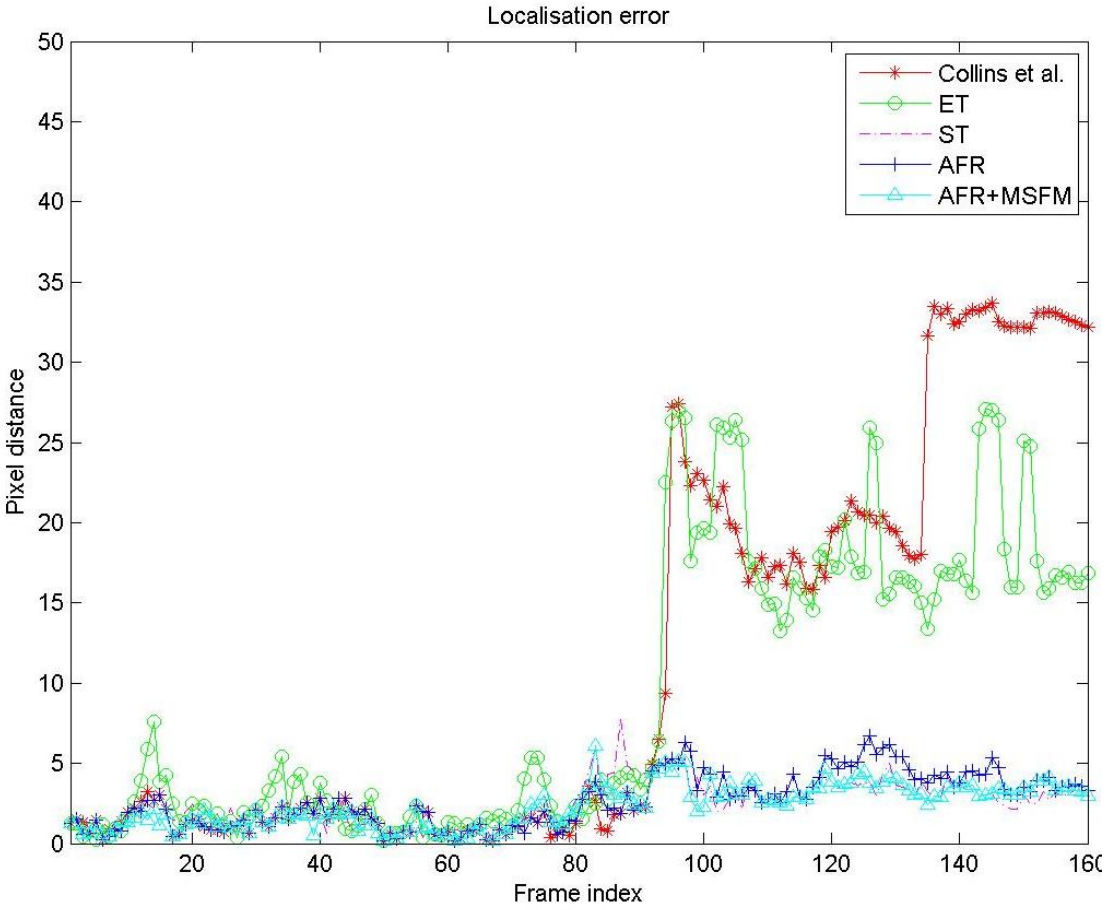


Figure 4.21: Scenario D, Example 1 localisation errors in pixel distance between each tracker and manually-labelled ground truth. Collins et al. and Ensemble Tracking (ET) were both distracted around frame 90, whereas the SemiBoost tracker (ST), AFR and AFR+MFSM all succeeded in tracking throughout the sequence.

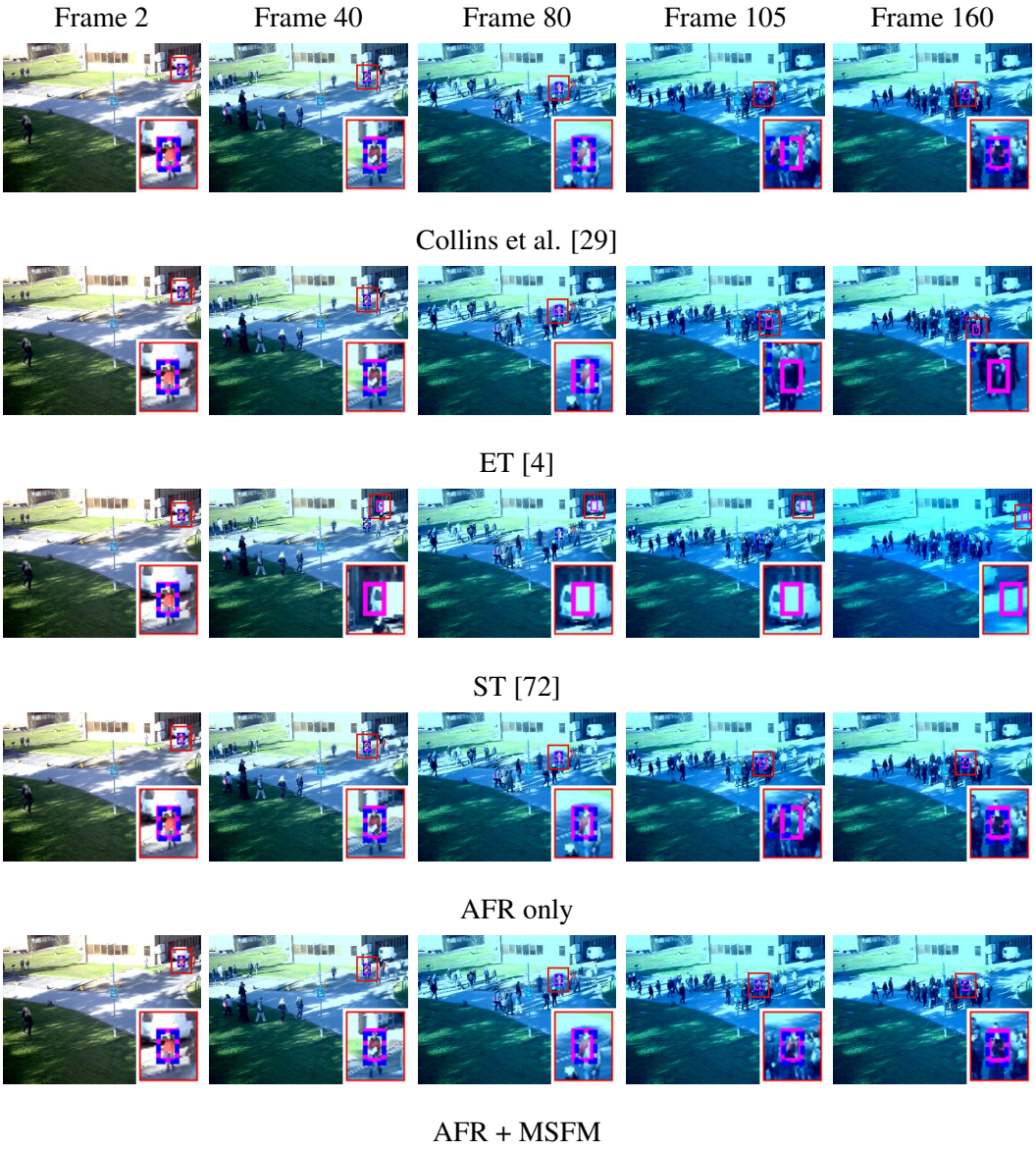


Figure 4.22: Scenario D, Example 2 Tracker Output: Tracking under occlusion and lighting change. The dashed dark blue box shows ground truth. The red and blue pixel values were modified (from left to right respectively) to 100%, 70%, 46%, 40% and 39% (for red) and 100%, 114%, 128%, 139% and 135% (for blue) of the original average pixel values. The SemiBoost tracker (ST) failed within ten frames while the others continued to track. Collins et al., Ensemble Tracking (ET) and AFR all became distracted around frame 95, with AFR recovering within fifteen frames and Collins et al. recovering within thirty frames. AFR+MSFM stayed locked onto the target for the duration of the sequence. See Figure 4.25 for the ground truth plot.

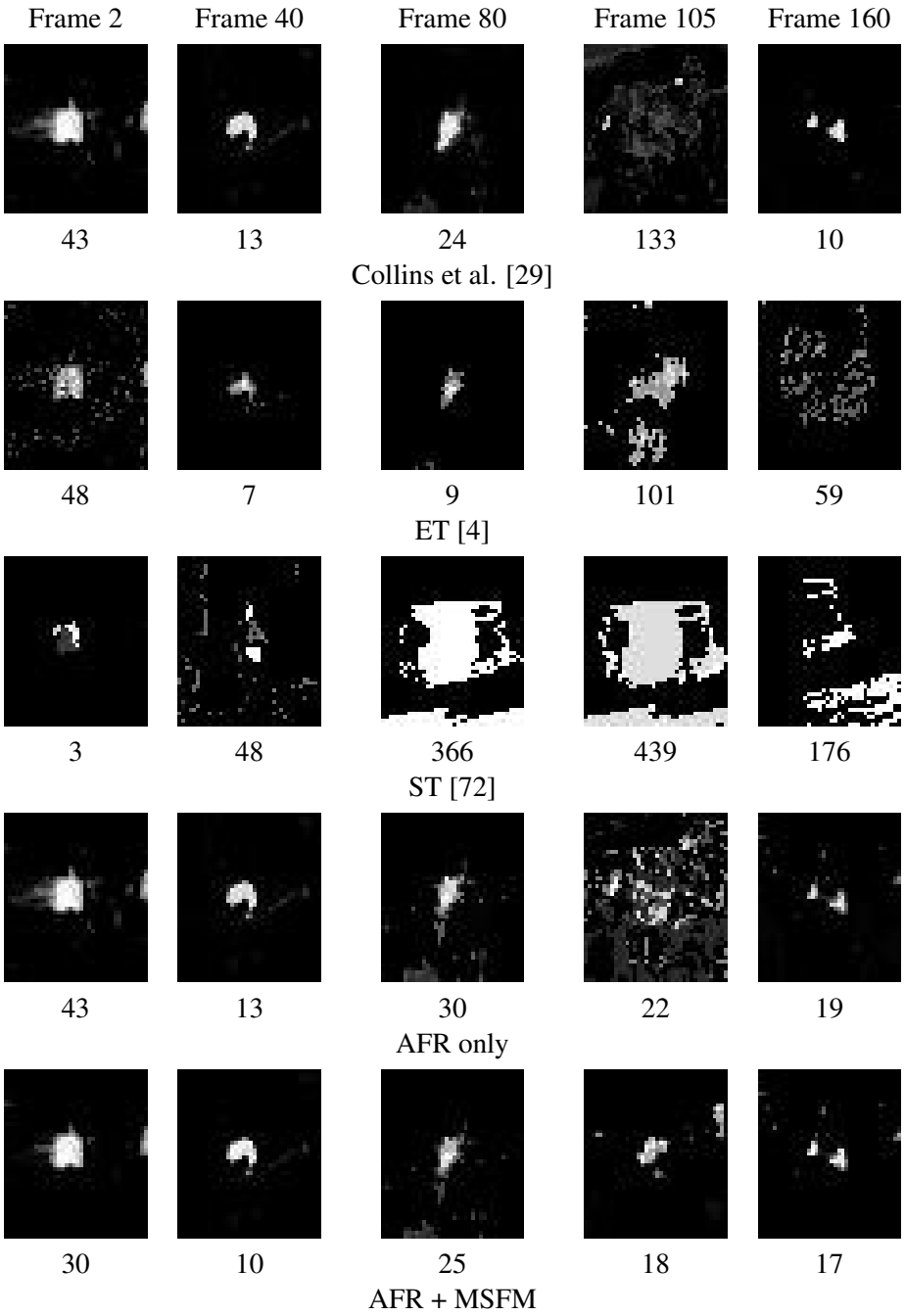


Figure 4.23: Scenario D, Example 2 Confidence Maps: The numbers below each image are the rounded sums of the confidences of pixels in the surround region. Due to early failure, the Semi-Boost tracker (ST) showed the least relevant confidence maps overall, with Ensemble Tracking (ET) also noisy following an unrecoverable failure around frame 95. As has been consistently demonstrated, MSFM helped to support AFR by reducing noise and improving tracking performance. Collins et al. showed a very similar quality in its confidence maps for this sequence although tracking performance was not up to the consistency of AFR and AFR+MSFM.

reference models.

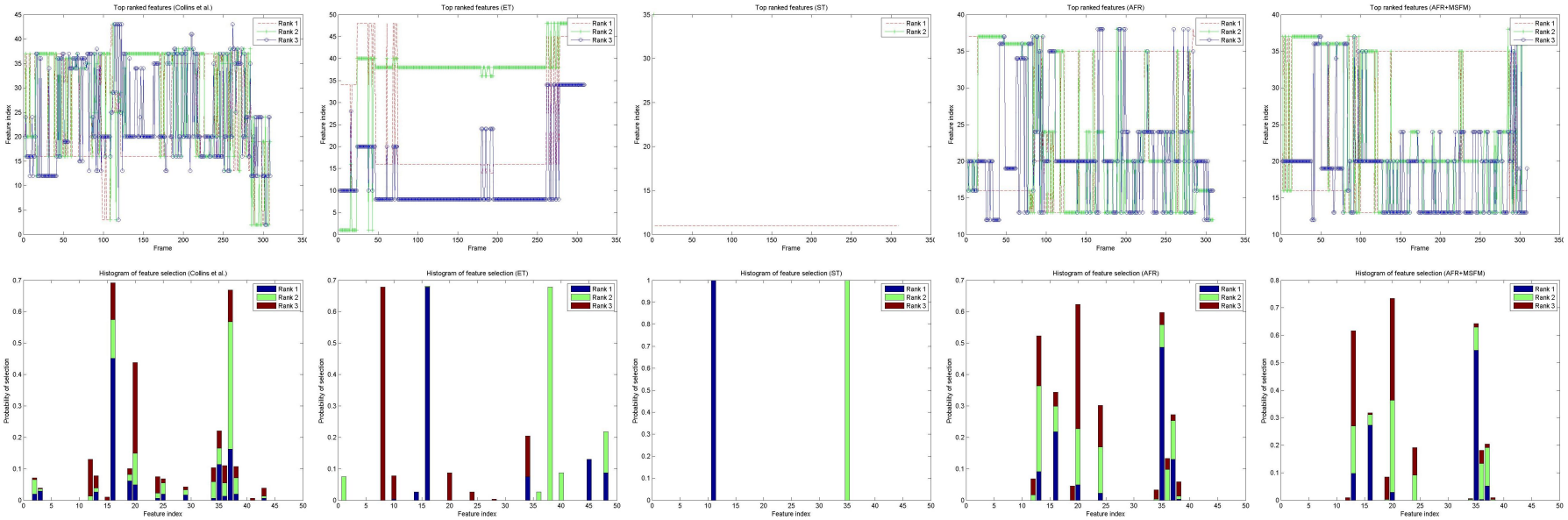
4.4 Discussion

The framework discussed in this chapter constitutes a general dynamic sampling strategy for tracking which is inherently adaptive spatially, temporally and *featurally*. Although the specific implementation we provide is in many ways rather simple, the different parts may be increased in sophistication without violating the overall ideology.

We employ canonical correlation analysis (CCA) as a method for estimating featural redundancy by computing the correlation between the data for different features. This has the advantage that distributions need not be estimated, avoiding the need for large amounts of data which are typically unavailable, especially in tracking situations such as those depicted for Scenario C (Figure 4.14). Furthermore, it is computationally much cheaper than most feature selection algorithms, an important factor in a time-critical application such as object tracking. However, the selection of feature subsets with low redundancy is a complex business and the subject of much intense research (see Section 2.1). Despite the experimental results given here, a further objective study of the "quality" of features recovered using our CCA-based attribute method would be desirable as well as the effects of modifying the weights for each of the attributes in the ranking process.

In our experiments we employed a rather limited set of features, all of which are photometric. They constitute a set of linear combinations of RGB features and had been used previously to demonstrate the idea of feature selection for tracking by Collins et al. [29]. Here, we again employed the same features for simplicity and to demonstrate our improved framework using the same feature pool. It would be interesting to conduct further experiments comprising larger feature pools incorporating geometric statistical features such as oriented gradients [41] and Local Binary Patterns [154] in addition to photometric features including alternative colour spaces.

Our framework includes the dynamic re-evaluation of the priors used for pixel classification (Equations 4.13 and 4.14) and updating of feature reference models (Equation 4.6). This is currently based on comparing the data for foreground and background across frames and the relative overlap between them as indicative of tracker errors or object occlusion. Further investigation should explore the limits of this approach and more sophisticated methods developed for controlling adaptation, including alternative mechanisms for distinguishing different causes of tracker



Collins et al. ET ST AFR AFR+MSFM

Figure 4.24: Scenario D, Example 2 feature selection statistics. Top row shows top three ranked (or weighted) features for each frame and the second row the corresponding frequencies of selection for each feature in the pool. As for all previous experiments, AFR+MSFM added further control to the feature ranking and selection statistics of AFR alone which in turn translated into actual performance (see Figure 4.25). The SemiBoost tracker (ST) again showed significant rigidity in feature ranking but failed early on. The statistics for Collins et al. and Ensemble Tracking (ET) showed a less haphazard characteristic than normal, but neither were able to track as consistently as AFR or AFR+MSFM.

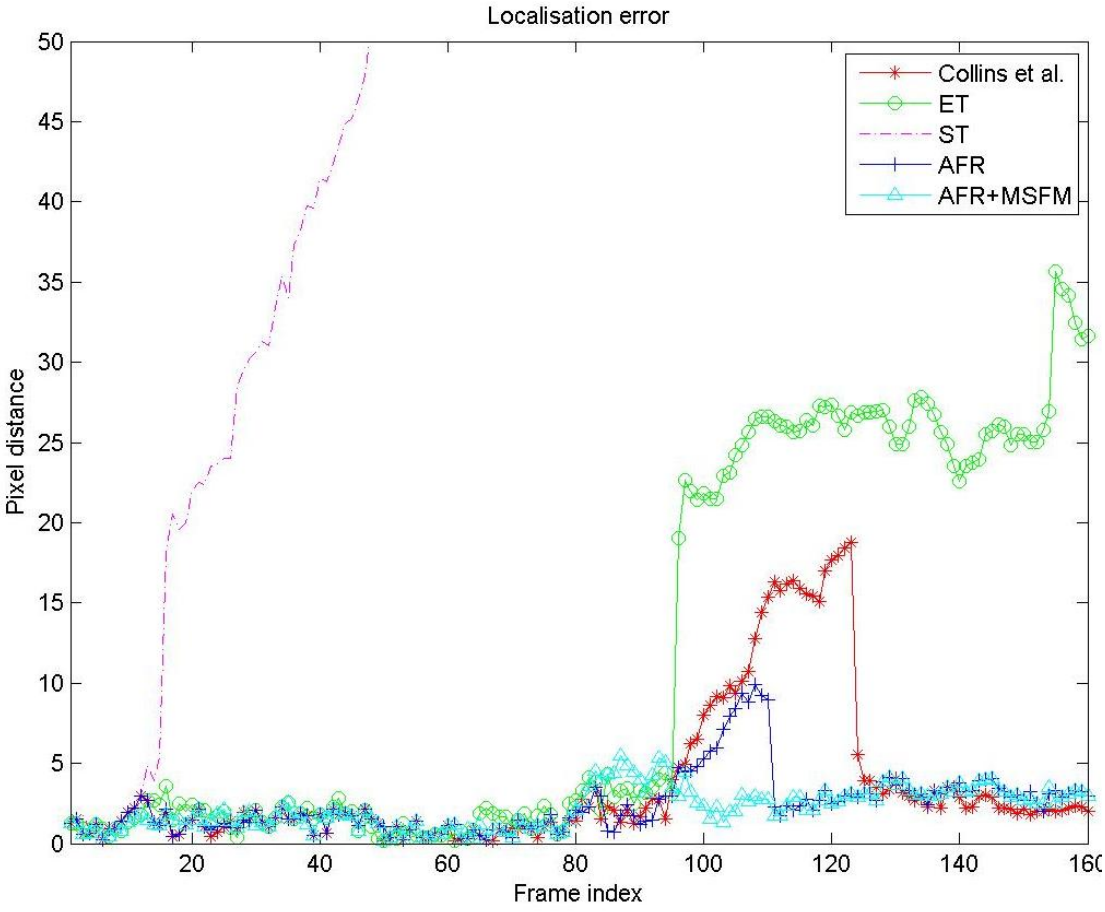


Figure 4.25: Scenario D, Example 2 localisation errors in pixel distance between each tracker and manually-labelled ground truth. The SemiBoost tracker (ST) fails very early, whereas Ensemble Tracking (ET) continues up to around frame 95 where it fails irreversibly. Both Collins et al. and AFR become distracted by the same proximal distractor around the same time, with the latter recovering sooner than the former. Only AFR+MSFM was able to maintain a lock for the duration of the sequence.

failure and their corresponding implications for reference model updates.

Currently, this framework focuses on the tracking of a single object. A natural way forward is to integrate data association techniques such as Joint Probabilistic Data Association Filters (JPDAFs) [60] to handle multiple tracked objects simultaneously in difficult scenarios and in a unified and rigorous manner.

4.5 Summary

In this chapter, we proposed a general dynamic spatial, temporal and featural sampling strategy for more robust tracking in difficult conditions. This involves combining the knowledge of trends in target object appearance over time with appearance characteristics in each frame to help overcome difficulties such as lighting change and partial occlusions. The benefit of this approach over previous methods are twofold:

1. Features are selected according to a measure of their correlation with other features in the pool in order to reduce redundancy among those selected at each frame

2. Feature reference models are *dynamic* rather than static and bypass unrealistic assumptions made by previous methods, in the process contributing significantly to the overall strategy of trading off short-term evidence with long-term trends

In implementing this strategy, we formulated a novel tracking framework called Adaptive Multi-feature Association (AMA) which comprises two methods for improving dynamic feature ranking and selection in tracking:

1. Attribute-based Feature Ranking (AFR) which combines two attribute measures, (a) a measure of discriminability and (b) a measure of independence to other features

2. Multiple Selectively-adaptive Feature Models (MSFM) which involves maintaining a dynamic feature reference of target object appearance. These are updated selectively against current image evidence for ranking features and classifying pixels in each frame

This framework implemented the proposed three-step filtering strategy comprising (a) focus-of-attention, (b) featural sample filtering and (c) spatial sample filtering.

We compared the performance of our framework to the established methods of Collins et al. [29], Avidan [4] and Grabner et al. [72] using four challenging real-world scenarios. We

showed in Scenario A that AFR selected more stable features which led to cleaner confidence maps that emphasised the most salient discriminative part of the tracking target object. Using AFR+MSFM improved pixel classification due to the up-to-date feature reference models and further improved the confidence maps. In Scenario B, we showed two examples of how these factors helped to overcome severe lighting changes which caused the Collins et. al, Ensemble Tracking and SemiBoost trackers to fail. Specifically, we demonstrated how the dynamic reference models enabled the tracker to adapt and recover in the face of a temporary and significant lack of saliency of the tracked object due to a severe lighting deficiency in Example 1 as well as preventing failure during a significant ambient colour change in Example 2. In Scenario C, we tested robustness under significant partial occlusion in a cluttered environment. AFR tracked longer than Collins et al., Ensemble Tracking or SemiBoost tracking but was perturbed by occlusion. We showed how AFR+MSFM overcame this through the use of MSFM to maintain the relevance of feature reference models and counteract the tendency of the frame-specific models to drift terminally under such conditions. Consequently, the target object was successfully tracked subsequent to reappearance from occlusion. In Scenario D we examined tracking under both difficulties of changing lighting and clutter/occlusion simultaneously, again with two examples; one of severe brightness change and the other of strong colour change. For the former, the SemiBoost, AFR and AFR+MFSM trackers were successful, whereas in the latter, only AFR+MFSM tracked throughout.

In these experiments, AFR was able to select more descriptive, appropriate features more consistently than Collins et al., Ensemble Tracking and SemiBoost tracking by virtue of the CCA-based Independence attribute. MSFM also consistently showed in every experiment that it was able to strongly support AFR in improving pixel classification by way of maintaining updated models which significantly reduced the chance of model drift, resulting in greater robustness in tracking. This suggests a more meaningful balance between flexibility and rigidity to obtain a better level of practical real-world consistency. Furthermore, the AMA framework is less complex than both Ensemble Tracking and SemiBoost tracking.

In the next chapter, we show how featural sampling may be employed as a natural and logical improvement to the use of well-known Local Binary Pattern methods [154]. Although these methods can be applied to a multitude of applications, we focus specifically on the use of LBP for recognition applications.

Chapter 5

Local Binary Pattern Feature Sampling for Recognition

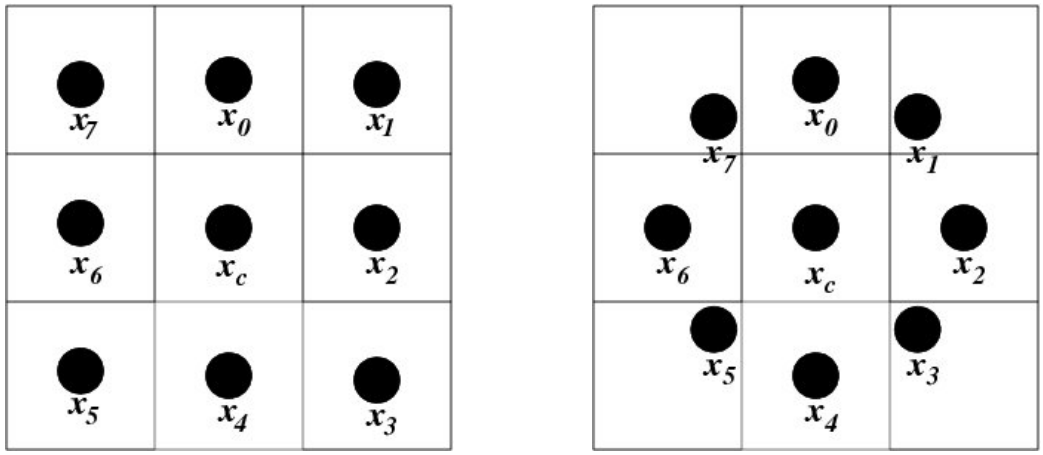
Visual recognition, like visual tracking, finds its basis in the problem of object association. Although tracking is usually concerned with distinguishing between the foreground object being tracked and the background, this can naturally extend to discriminating between different foreground objects under difficult conditions, e.g. as addressed by Song et al. [204] who employed object classification to perform disambiguation when tracking people in crowded scenes. The most appropriate features to use can be highly context dependent; in the same way that effective classification-based tracking will employ the most salient features at any given time to reliably perform discrimination between a target object and the background or other foreground objects present at that time, effective recognition strategies will employ the most salient features for reliably classifying objects as one of many possible classes. In either case, there is a fundamental requirement for a sampling strategy that scrutinises and selects the most appropriate parts or features of an object for improved robustness (in terms of classification error) and efficiency (the amount of computation required for discrimination).

In this chapter we address some of the limitations of one specific type of methodology for object discrimination, namely Local Binary Pattern (LBP) based recognition applications. LBP methods have been used extensively for a huge range of applications, including texture discrimination [130, 155] demonstrating excellent results and good robustness against rotation and global illumination changes, texture segmentation [172] and recognition of facial identity [2] and expression [53, 193]. This paradigm involves representing classes of objects such as faces or textures by the joint statistical modelling of Boolean features yielded by thresholding samples from

the surround of each pixel in corresponding images to capture local structure. We describe the limitations of previous LBP methodologies and argue that they are borne from a fundamentally inflexible approach to modelling LBP statistics which: (a) either limit the spatial area over which models may capture information or otherwise average over potentially useful details; (b) incorporate possibly redundant information which wastes resources; (c) decouples statistics in an ad hoc manner; and (d) builds models in a spatially non-selective, context-agnostic way which further impacts on classification accuracy and betrays the inherent importance of adaptive visual sampling approaches to discrimination. We then propose a framework which solves all of these problems and may be added without modification to many existing LBP-type methods which involve modelling distributions of jointly encoded binary sequences such as [127, 118]. This framework consists of: (a) A novel feature selection algorithm designed for binary data, called Binary Histogram Intersection Minimisation (BHIM), which is capable of finding stronger, less-redundant feature subsets than two state-of-the-art algorithms for binary feature selection; (b) The encoding of selected features to form distributions from context-dependent spatial topologies called Multiscale Selected Local Binary Features (MSLBF); and (c) The use of MSLBF models in a pairwise-coupling [82] scheme to enable the most appropriate samples to be used depending on the two classes being compared.

5.1 Scope of the problem

Local Binary Pattern methods are concerned with the statistics of jointly encoded features which reflect local intensity fluctuations around each pixel in an image. These features are formed from Boolean values (known as *textons*) derived from sampling intensities surrounding each pixel, thresholding them by the central intensity and noting the sign of the result. The surround may be arranged as square or circular. Figure 5.1(a) illustrates the square-surround approach which simply involves treating each pixel surrounding the centre of a 3x3 centre-surround arrangement as a sample for thresholding, leading to operators labelled as LBP_P with P denoting the number of textons. Figure 5.1(b) shows a circular approach which enables rotational invariance [155] but involves computing sub-pixel intensities prior to thresholding. These operators are labelled $LBP_{P,R}$ with R referring to the radius at which the P samples are taken. In either case, the resulting n textons $\{t_0, t_1, ..., t_{n-1}\}$ are (in a fixed-sequence) interpreted as a binary number and encoded by the corresponding decimal equivalent w:



(a) (b)

Figure 5.1: Strategies for LBP surround sampling. (a) The intensity samples used for thresholding are simply the pixels $x_0, ..., x_7$ from the 3x3 grid surrounding the centre pixel x_c. (b) Samples are taken by sub-pixel sampling on a circle at a fixed distance before thresholding.

$$w = \sum_{p=0}^{n-1} t_p 2^p, \;\; t_p = \begin{cases} 1, & (x_p - x_c) > 0 \\ 0, & otherwise \end{cases} \tag{5.1}$$

where x_c denotes the intensity of the centre pixel.

Each texton may be viewed as a individual feature. As such, they form slices in the Spatial-Featural-Volume (SFV) for an LBP processed image (see Figure 2.21). Each slice then constitutes a binary image with each pixel represented by the Boolean value for the corresponding texton relative to that pixel. The LBP histogram for an image is then a distribution over the decimal values $w_1, w_2, ..., w_m$ for all m pixels in the image, defining the *joint* statistics of all of these SFV slices. The histograms for all images in the class are averaged and used as a descriptor for that class. During a recognition task, the histogram for a single image may be generated and the best match with the various models found using a histogram comparison method such as histogram intersection or Kullback-Leibler divergence.

In order to encompass larger support areas, the notion of *multipredicate LBPs* [130] involve several circular sampling arrangements at fixed radii centred on each pixel, with histograms generated separately for each radius (see Figure 5.2). Each increasing radius comprises a larger number of samples for thresholding and encoding. Here, the resulting histograms (predicates) are appended together to form a single descriptor for the class.

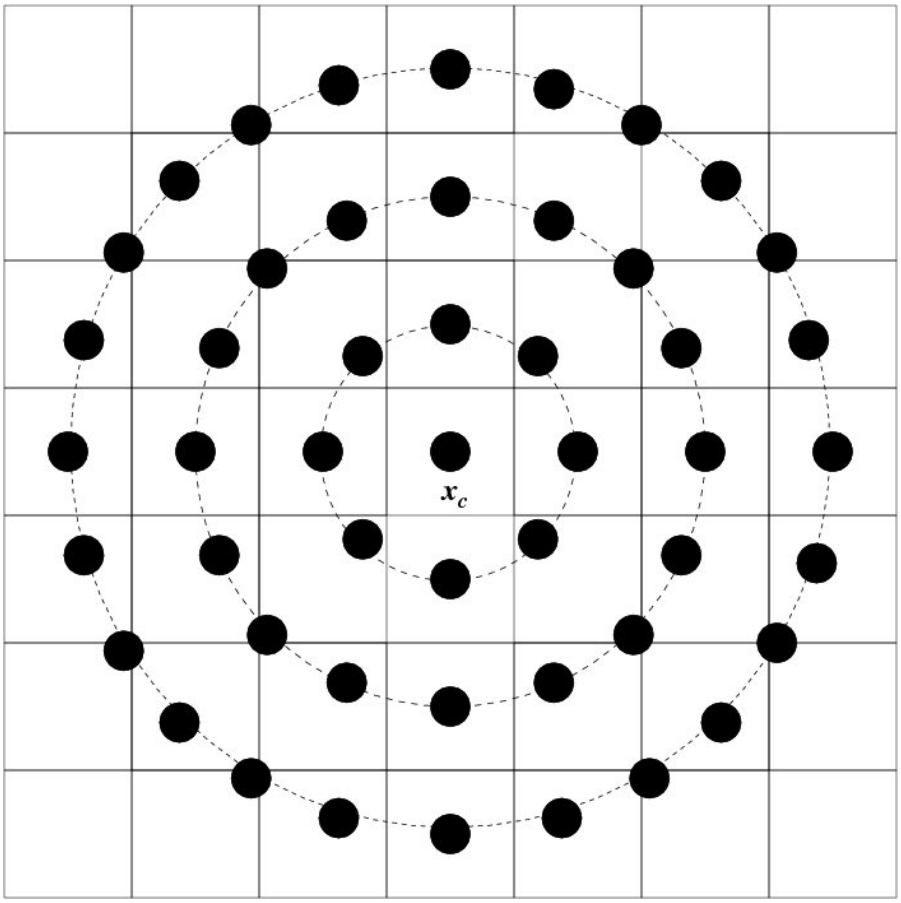


Figure 5.2: An example of a fixed pre-determined sampling strategy for a multipredicate LBP. Each radius represents a spatial arrangement from which image samples are taken and corresponding textons derived for encoding (see Equation 5.1). The dashed lines indicate the samples which are considered jointly. As such, each radius is considered separately from the others, with the result that all scales are statistically disjoint from each other.

There are several drawbacks to this scheme:

1. For circular neighbourhoods with large radii (scales), the number of samples required for a reasonable characterisation of local structure may not in practice be amenable to joint statistical modelling, since the resulting histograms may be very large. This puts great emphasis on the number of data required to adequately estimate a distribution;

2. For the same reason, the number of scales that may be used are limited which restricts the spatial support area for capturing structure at each pixel's locale. A common multipredicate topology employed is 8 samples at a radius of 1 pixel from the centre, 16 samples at radius 3 and 24 samples at radius 5. Even for this modest range of spatial support, the resulting full appended class histogram will contain millions of bins;

3. The samples at each radius are modelled disjointly from the others. This introduces an unnatural decoupling of statistics and prevents potentially important inter-scale relationships from being captured in the resulting models;

4. Encoding *all* individual surround features at a certain radius may introduce redundancy into the representation, since two or more textons at a radius may be strongly correlated with each other. Given already limited resources this is highly undesirable.

Efforts have been made to reduce the size of histograms as well as to improve the relevance of patterns used for models. Experiments by Ojala et al. [155] showed that certain patterns correlated well with real-world structures and helped improve performance as well as significantly trim the size of histograms. Those binary sequences with at most two zero-to-one or one-to-zero transitions, called "uniform" patterns, were found to be most useful. Histograms are then formed only from the encoded decimal values for uniform patterns with a single bin for all that are non-uniform. Further, all patterns may be rotated to a canonical position which enables rotation invariance and further reduces the number of patterns that need be catered for. The resulting operators are labelled as $LBP^{riu2}_{P,R}$, with superscripts *ri* denoting rotation invariance and $u2$ uniform patterns. Other work geared towards improving the selectivity of models include Lahdenoja [107] who exploited symmetry to reduce feature vector lengths and improve efficiency, Shan et al. [193] who employed boosting techniques on LBP classifiers to improve facial expression discrimination and Liao et al. [117] who, rather than use the heuristically-inspired uniform patterns, trimmed the bins with the lowest value from a histogram of all possible patterns thereby making

the process more contextually relevant. However, all of these methods involve analysing patterns derived from all textons *jointly*. As such, they do not deal with the inevitable redundancies that may be found within structures extracted from the fixed circular topologies.

To increase spatial support, cellular automata were used by Mäenpää and Pietikäinen [127] to encode multiple scales (radii) with each scale corresponding to a time-step in a cellular automaton. With this scheme, an arbitrary number of scales can be incorporated into a single descriptor comprising the marginal distributions for each scale and a cellular automaton rule to bind them together. However, this approach requires the same number of samples for each scale, resulting in sparser sampling at the largest scales. They countered this by performing low-pass filtering at each sample point for the larger scales in order to encompass information over a larger area; however, this serves to smooth over any fine detail that may be relevant. Furthermore, this again does not address any redundancy of information incorporated into patterns at each scale.

5.2 Multiscale Selected Local Binary Features

All previous methods involve working with textons derived from fixed sampling strategies; that is, they jointly encode *all* of the textons from a circular neighbourhood. As explained above, this in many cases will incorporate redundancy into the information. Furthermore, practicalities ensure that the number of scales included are limited with larger radii further limited in the number of samples they may incorporate as well as forcing the statistical decoupling of individual scales. In terms of the Spatial-Featural-Volume (Figure 2.21), previous methods for reducing valid patterns or selecting the most useful ones amount to a fixed pre-determined sampling strategy which performs filtering within the *spatial* domain, with patterns from individual pixels being accepted or rejected according to some criterion of acceptability. We argue that a more natural approach to improving the power of texton features and one that solves all of the problems described in Section 5.1 is to *learn* a sampling strategy that selects a sparse set of the most useful textons to focus attention on, i.e. filtering within the *featural* domain. The *selected* textons may then be encoded in the same way as a traditional LBP predicate (see Figure 5.3). Furthermore, the learned sampling strategies should be *adaptive*. The most efficient subset of textons for discrimination may be different according to context, i.e. depending on the objects being compared. The benefits of this approach are listed thus:

1. Any number of samples may be collected from any number of scales and incorporated into

a pool of textons for the selection process;

2. The selected textons may be drawn from any spatial position and from any scale and modelled jointly. Consequently, spatial support areas can be arbitrarily increased by adding as many scales are required;

3. The resulting models are fully statistically coupled across scales;

4. The models are sparse, compact and highly descriptive by virtue of the feature selection process, increasing discrimination power as well as computational efficiency;

5. The new selection-based framework may be seamlessly appended to many previous improvements to traditional LBP methodologies, such as the use of low-pass filtering to integrate over larger areas [127], the multi-block approach for effectively subsampling images [118] or further histogram trimming based on bin frequencies [117].

We propose a novel algorithm called Binary Histogram Intersection Minimisation (BHIM) for performing the feature selection process. This is a general filter method designed specifically to find strong binary features for two-class classification tasks. This is used to select textons from a pool which may comprise any number of textons across any number of scales. The chosen textons may then be jointly encoded as per the usual fashion of computing the decimal equivalent of the corresponding binary sequence.

5.2.1 Texton selection by Binary Histogram Intersection Minimisation

There is a large body of work dedicated to the theory and practicalities of feature selection. This field of research is concerned with the identification of subsets of variables from training data which are good predictors of the class variable. Theoretically, the best subset $\mathbf{M}$ of a set of variables $\mathbf{X}$ may be considered as that which renders the class variable Y as *conditionally independent* of the set difference between $\mathbf{X}$ and $\mathbf{M}$ given $\mathbf{M}$ (Pearl [162]):

$$Y \perp\!\!\!\perp (\mathbf{X} \backslash \mathbf{M}) | \mathbf{M} \tag{5.2}$$

The subset $\mathbf{M}$ is known as a *Markov blanket*. It has been shown to potentially constitute the set of *strongly relevant* (see Section 2.1.1) features (Tsamardinos et al. [225]). Another perspective may be provided by the notion that the optimal subset $\hat{\mathbf{M}}$ minimises the conditional entropy $H(Y|\hat{\mathbf{M}})$ between Y and features in $\hat{\mathbf{M}}$ taken jointly (Fleuret [55]):

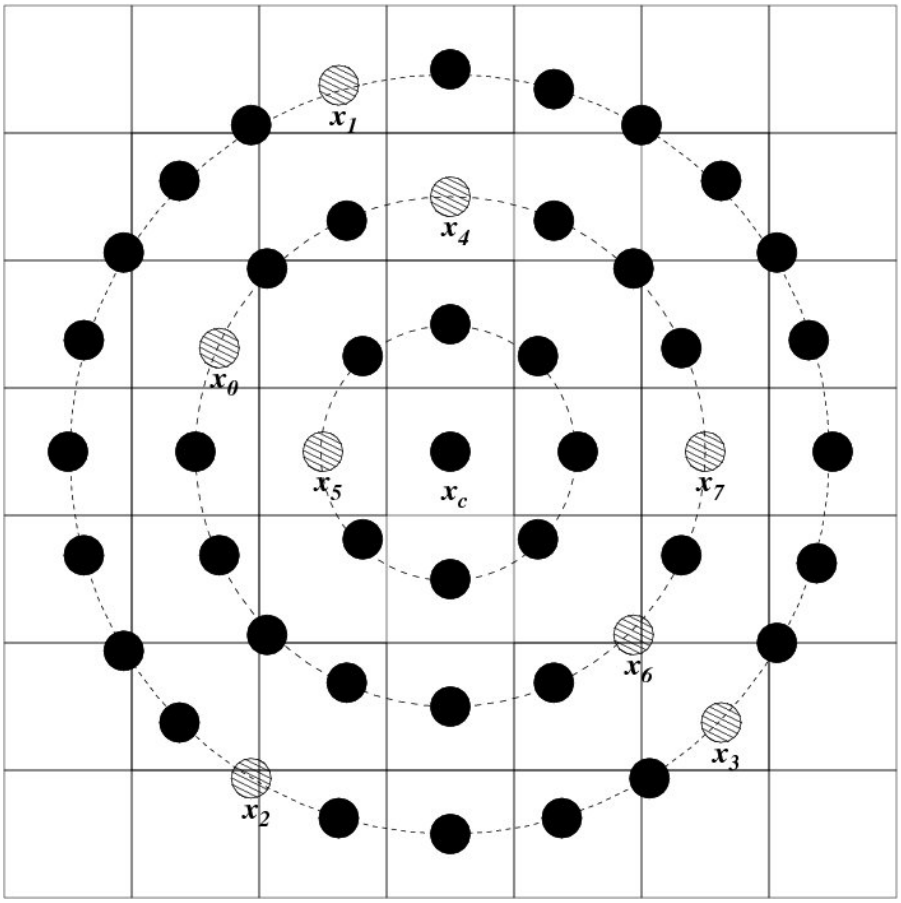


Figure 5.3: An example of a learned sampling strategy for a Multiscale Selected Local Binary Features (MSLBF) operator. A feature selection step chooses the most useful textons which are then used to form sparse, highly discriminative models with distinct spatial topologies which are fully coupled across scales. This figure shows three scales although any number of scales and corresponding samples may be added to a pool of textons for selection.

$$\hat{\mathbf{M}} = \arg\max_{\mathbf{M}} \{H(Y|\mathbf{M}) | \mathbf{M} \in 2^{\mathbf{X}}\} \tag{5.3}$$

where $\hat{\mathbf{M}} = \{X_{b_1}, X_{b_2}, ..., X_{b_K}\}$ and b_k denotes the index of the k'th feature selected from $\mathbf{X}$.

Most feature selection approaches do not attempt to recover the Markov blanket since it is usually impractical to test all possible subsets of variables, particularly when $|\mathbf{X}|$ is large. Those algorithms which have been proposed tend to fall into either the forward selection or backward elimination categories (see Section 2.1.2) with some hybrid variants [225]. Two established algorithms that are well-suited to dealing with binary variables are the AdaBoost (Adaptive Boosting) algorithm (Freund and Schapire [61]) which may be used for feature selection and the more recent Conditional Mutual Information Maximisation (CMIM) (Fleuret [55]) technique. AdaBoost is a wrapper method which involves the training of multiple weak classifiers on weighted training sets which are reweighted at each step such that successive classifiers focus more strongly on the samples that are previously incorrectly classified. Its implementation for binary training data is particularly efficient (see Figure 2.1). CMIM is a filter method which is theoretically more capable of determining stronger features because of its consideration of features in *combination* rather than individually; that is, each feature is ranked according to its relationship to both the class variable and to features already selected. It employs the information-theoretical idea of *conditional mutual information*:

$$I(U;V|W) = H(U|W) - H(U|W,V) = \int_W \int_V \int_U p(U,V,W) log \left\{ \frac{p(W)p(U,V,W)}{p(U,W)p(V,W)} \right\} \tag{5.4}$$

where U, V and W are random variables, $I(U;V|W)$ denotes the mutual information between U and V conditional on W and $H(U|W)$ is the entropy of U conditional on W. More specifically, CMIM selects candidate features X_j which maximise their mutual information with the class variable Y *conditional on the features already picked* $\{X_{b_1}, ...X_{b_k}\}$. However, since doing this would require the prohibitive step of estimating large joint densities (i.e. $I(Y;X_j|X_{b_1}, ..., X_{b_k})$), a trade-off is made which involves selecting the feature X_j which maximises its mutual information with Y conditional on the feature from the set already picked that *minimises* its mutual information with X_j:

$$b_1 = \arg\max_j I(Y;X_j), \quad b_{k+1} = \arg\max_j \left\{ \min_{l \leq k} \{I(Y;X_j|X_{b_l})\} \right\} \tag{5.5}$$

This forms an efficient, suboptimal trade-off which at most requires estimating distributions of joint triplets of variables. Moreover, efficiency can be further improved when working with binary data as the joint distributions can be estimated by simple summation operations. The algorithm has been shown to improve performance over other methods such as AdaBoost (see Fleuret [55]).

Here, we devise an algorithm specifically tailored for binary data known as Binary Histogram Intersection Minimisation (BHIM). Like CMIM and AdaBoost, it follows a greedy feed-forward procedure which ensures that it is suboptimal. Unlike the linear-time CMIM and AdaBoost algorithms, it is at worst exponential-time. However, it offers the following benefits:

1. It considers *all* currently selected features jointly when considering a new candidate as opposed to a single one as for CMIM;

2. Despite being exponential-time, the algorithm in most cases does not require prohibitive computational time to function because it is naturally restricted by the limits of the training data;

3. Strong features require fewer computational resources to determine than weak ones, making the algorithm quite adaptive in terms of required resources;

4. The computationally-cheap histogram intersection is employed at each iteration to compute the divergence of histograms. Furthermore, these are only ever computed on two-bin binary histograms. Entropy computations are unnecessary;

5. Despite the use of histogram intersection, the algorithm tends to lead to a lower entropy for the class variable conditional on the chosen subset than those chosen by AdaBoost or CMIM, implying a more descriptively potent subset of features.

The BHIM algorithm functions as follows: when provided with two binary data sets, the algorithm attempts to find K binary features from the total feature pool whose joint distributions for each of the two models are strongly divergent. More precisely, given two classes with datasets $\mathcal{P}$ and $\mathcal{Q}$ constructed from random variables X_j corresponding to binary features indexed by j, $1 \leq j \leq J$, the objective of the algorithm is to find a set $B = \{b_1, b_2, ..., b_K\}$ where the b_ks are the indices of the selected features from the feature pool. Histogram intersection is employed in the scoring of features. These are only ever computed with binary histograms and so

the "histogram distance" $[1 - HI\{h(X|\mathcal{P}), h(X|\mathcal{Q})\}]$ given normalised histograms $h(X|\mathcal{P})$ and $h(X|\mathcal{Q})$ for a binary feature X generated from datasets $\mathcal{P}$ and $\mathcal{Q}$ is computed more simply as $|h(X=1|\mathcal{P}) - h(X=1|\mathcal{Q})|$. Each b_k is selected as follows:

$$b_1 = \underset{j}{arg\,max} \ |h(X_j = 1|\mathcal{P}) - h(X_j = 1|\mathcal{Q})| \tag{5.6}$$

$$b_{k+1} = \underset{j \notin B'}{arg\,max} \sum_{\mathbf{X} \leftarrow \{0,1\}^k} S(\mathbf{X}, B', j) \tag{5.7}$$

where $B' = \{b_1, b_2, ..., b_k\}$, $k < K$ is the partial set of features selected so far and

$$S(\mathbf{X}, B', j) = P(\mathbf{X}_{B'}|\mathcal{P}) \cdot P(\mathbf{X}_{B'}|\mathcal{Q}) \cdot |h(X_j = 1|\mathbf{X}_{B'}, \mathcal{P}) - h(X_j = 1|\mathbf{X}_{B'}, \mathcal{Q})| \tag{5.8}$$

The terms $P(\mathbf{X}_{B'}|\mathcal{P})$ and $P(\mathbf{X}_{B'}|\mathcal{Q})$, where $\mathbf{X}_{B'} = \{X_{b_1}, ..., X_{b_k}\}$, are the joint probabilities of occurrence of a specific instance of the binary vector $\mathbf{X}$ over the k previously selected features with indices $B' = \{b_1, b_2, ..., b_k\}$, for classes $\mathcal{P}$ and $\mathcal{Q}$ respectively. Similarly, the terms $h(X_j = 1|\mathbf{X}_{B'}, \mathcal{P})$ and $h(X_j = 1|\mathbf{X}_{B'}, \mathcal{Q})$ are the normalised binary histograms for feature X_j conditional on the specific binary vector $\mathbf{X}$ over the selected features B'.

At step $(k+1)$, the algorithm computes Equation 5.7 which, for each feature $j \notin B'$, includes the expectation of the conditional distribution histogram distance between the datasets $\mathcal{P}$ and $\mathcal{Q}$ over the joint distribution for feature set B'. This involves at most 2^k values for $\mathbf{X}$, although only values present in both datasets need be included in the computation. Datasets containing features with strongly separating statistics will generally have far fewer shared values between them, which results in those features requiring less computation time to identify. The feature $j \notin B'$ that maximises the expectation is chosen for adding to the currently chosen subset B'. The algorithm stops when K features have been selected or no value of $\mathbf{X}$ has a positive probability for both models simultaneously, meaning their corresponding joint binary histograms have zero intersection. Algorithm 5.1 describes BHIM in pseudocode.

This algorithm may be implemented efficiently by associating each sample with a decimal value encoding the patterns derived from the currently selected feature subset and updating these values after each selection. A unique set of the shared patterns (decimal values) between classes may be easily assembled and for each pattern in the set and the corresponding samples indexed accordingly for each class. Note that at each iteration and for each new feature candidate, the

Input: Two datasets $\mathcal{P}$ and $\mathcal{Q}$; generated with J features; a number K of features to select

Output: A set $B = \{b_1, ..., b_K\}$ of at most K selected features

$B = \{b_1\}$;

for $k = 2$ *to* K **do**

  $\mathbf{w}_{\mathcal{P}} = decval(\mathcal{P}, B)$;

  $\mathbf{w}_{\mathcal{Q}} = decval(\mathcal{Q}, B)$;

  $\mathbf{W} = \mathbf{w}_{\mathcal{P}} \cap \mathbf{w}_{\mathcal{Q}}$;

  if $\mathbf{W} = \{\}$ **then**

    return B;

  end

  Set all scores $\mathbf{s}_j = 0$, $j = 1..J$;

  for $j = 1$ *to* J, $j \notin B$ **do**

    for *each* $\mathbf{w} \in \mathbf{W}$ **do**

      $\mathbf{s}_j = \mathbf{s}_j + S(\mathbf{w}, \mathcal{P}, \mathcal{Q}, B, j)$;

    end

  end

  $b_k = \arg\max_j \{\mathbf{s}_j\}$, $j = 1..J$;

  $B = B \cup \{b_k\}$;

end

Algorithm 5.1: The Binary Histogram Intersection Minimisation (BHIM) algorithm for selecting K features from two binary data sets $\mathcal{P}$ and $\mathcal{Q}$ containing J features. The function $decval(\zeta, B)$ returns a set of decimal values for each training sample in class ζ generated from the joint values across the features in B. The function $S(\mathbf{w}, \mathcal{P}, \mathcal{Q}, B, j)$ computes the weighted binary histogram distance corresponding to Equation 5.8.

class-conditional histograms that are compared represent distributions conditional on *all* features selected so far, as opposed to CMIM which only considers one of the already-selected features. These conditional distributions need only be formed for each shared pattern from the correspondingly indexed samples. Since these shared patterns represent the overlap between the classes given the chosen feature subset, at each iteration the feature candidates chosen are those that maximally reduce the overlap.

5.2.2 MSLBF classification procedure

The fewer the number of classes in a discrimination task, the less the potency required for individual features in discriminating between them. Arguably, this is particularly true of binary features taken in isolation due to their limited range of values. Consequently, larger and larger numbers of features may be jointly required for discrimination tasks with a large number of classes, such as face recognition. Moreover, different subsets of features may be the most appropriate for different subsets of classes, which also has the benefit of permitting more compact models. This arguably justifies a context-dependent approach to sampling. Here, to simplify the feature selection process and facilitate more compact and descriptive context-dependent models, we recast the N-class LBP histogram matching procedure as a pairwise-coupled approach [82], with a single N-class classification replaced by all possible 2-class classifications taken from the N classes. Each unique pair of classes is associated with its own selected features. This arrangement may be viewed as a context-dependent spatial and featural sampling strategy.

More specifically, an MSLBF classifier is simply a list of pairs of histograms, each pair uniquely corresponding to a specific two of N classes. Consequently there are $\frac{1}{2}N(N-1)$ binary classifiers required for an N-class classification task. Each classifier $c_{\mathcal{P},\mathcal{Q}}$ comprises a set of K selected feature indices $B_{\mathcal{P},\mathcal{Q}}$ corresponding to their positions in the feature pool along with two 2^K-bin histograms corresponding to the joint distributions over $B_{\mathcal{P},\mathcal{Q}}$, one for each of the two classes $\mathcal{P}$ and $\mathcal{Q}$. Given a set of training classes ζ_1 to ζ_N, the trainer cycles through all possible combinations of pairs of classes $\zeta_\mathcal{P}$ and $\zeta_\mathcal{Q}$, $\mathcal{P} \neq \mathcal{Q}$ and calls the feature selection algorithm with the samples for those classes to generate $B_{\mathcal{P},\mathcal{Q}}$. Adding classes is straightforward and requires N extra binary classifiers to be generated, one for each of the N classes against the new class indexed $N+1$. Each class n ($1 \leq n \leq N$) in an N-class problem has $N-1$ binary classifiers for comparing against each of the other $N-1$ classes.

Classification of an input involves keeping a score for each of the N classes. Since each

specific pair of classes has a separate set of discriminative features, histograms are assembled for each of the $\frac{1}{2}N(N-1)$ binary classifiers $c_{\mathcal{P},\mathcal{Q}}$ given their corresponding selected features. Each pair-specific input histogram is intersected with the two pair-specific model histograms, with the intersection value being added to the score of the class with the highest match. After all binary classifications are performed, the class with the highest score is assigned to the input. Algorithm 5.2 provides pseudocode for the classification procedure.

Input: Input data I to classify; $\frac{1}{2}N(N-1)$ classifiers $c_{\mathcal{P},\mathcal{Q}}$ for each unique pair $\mathcal{P}$ and $\mathcal{Q}$ of N classes, $\mathcal{P} \neq \mathcal{Q}$ each with corresponding selected features $B_{\mathcal{P},\mathcal{Q}}$ and associated histograms $h_{\mathcal{P},\mathcal{Q}}(X|\mathcal{P})$ and $h_{\mathcal{P},\mathcal{Q}}(X|\mathcal{Q})$

Output: Strongest matching class R for assignment to input data I

Set all scores $\mathbf{s}_j = 0,\ j = 1..N$;

for $\mathcal{P} = 1$ *to* $N-1$ **do**

  for $\mathcal{Q} = \mathcal{P}+1$ *to* N **do**

    $r = genhist(I, B_{\mathcal{P},\mathcal{Q}})$;

    $v_{\mathcal{P}} = HI\{r, h_{\mathcal{P},\mathcal{Q}}(X|\mathcal{P})\}$;

    $v_{\mathcal{Q}} = HI\{r, h_{\mathcal{P},\mathcal{Q}}(X|\mathcal{Q})\}$;

    if $v_{\mathcal{P}} > v_{\mathcal{Q}}$ **then**

      $\mathbf{s}_{\mathcal{P}} = \mathbf{s}_{\mathcal{P}} + v_{\mathcal{P}}$;

    end

    if $v_{\mathcal{Q}} > v_{\mathcal{P}}$ **then**

      $\mathbf{s}_{\mathcal{Q}} = \mathbf{s}_{\mathcal{Q}} + v_{\mathcal{Q}}$;

    end

  end

end

$R = \arg\max_j \{\mathbf{s}_j\},\ j = 1..N$;

Algorithm 5.2: The Multiscale Selected Local Binary Features (MSLBF) classification algorithm. $genhist(I, B_{\mathcal{P},\mathcal{Q}})$ is a function returning the histogram for input data I given the features $B_{\mathcal{P},\mathcal{Q}}$ specific to the pair of classes $\mathcal{P}$ and $\mathcal{Q}$. The $HI\{h(X|\Phi), h(X|\Xi)\}$ function computes the histogram intersection between two histograms $h(X|\Phi)$ and $h(X|\Xi)$.

5.3 Experiments

For the experiments, the aim was twofold:

1. Investigate the improvement in performance that can be gained from modelling jointly

across multiple scales spanning larger spatial support areas with many textons in the feature pool, as opposed to the limits on traditional LBP

2. Compare the three feature selection algorithms for the strength of features selected

As such, we present three experiments. Firstly, we focus on a classification task for textures, in particular the Outex database (Ojala et al. [153]). In doing so, we compare the performance of LBP and MSLBF classifiers as well as the quality of features selected for the MSLBF classifiers by three feature selection algorithms; namely, AdaBoost, CMIM and BHIM. Secondly, we apply our approach to face recognition on the challenging ORL face database (Samaria and Harter [186]). Similarly, we here compare the two types of classifier and the quality of MSLBF classifiers as derived from the use of the three feature selection algorithms. Both the texture and face experiments involved training MSLBF on larger feature pools encompassing more texton features and larger spatial support areas. Finally, we compare the performance of BHIM with CMIM and AdaBoost for selecting strong features from random simulated pairs of datasets. These pairs of datasets have randomly distributed joint distributions "embedded" at randomly selected feature positions with random degrees of overlap between the two classes. These form a target set of distinguishing features for the selection algorithms to find.

For each pixel of an image class, a sample was collected by considering several circular neighbourhoods at different radii centred on that pixel and collecting sub-pixel intensity samples at each radius before thresholding them by the intensity of the centre pixel itself. The resulting textons formed the features in the pool. Each feature selection algorithm was employed to select features specific to each pair of classes and form corresponding MSLBF classifiers. In addition, a traditional LBP classifier was also trained on both the texture data and the face data for comparison with MSLBF in classification. These were formed from three circular neighbourhoods at 1, 3 and 5 pixels radius and 8, 16 and 24 textons respectively.

5.3.1 Texture recognition

The MSLBF approach was applied to a suite of the Outex [153] database, specifically, Outex_TC_00000. This data set comprises 24 texture classes across 480 128×128 pixel images with 20 images per class. Samples are shown in Figure 5.4. An $LBP^{riu2}_{8,1} + ^{riu2}_{16,3} + ^{riu2}_{24,5}$ classifier was trained on part of the data (with samples defined by problem no.25 in the Outex_00000 suite) along with three MSLBF classifiers, each with 8 features selected per class-pair by a different

selection algorithm. These MSLBF classifiers were provided with more training data by including predicates constructed from circular neighbourhoods at 1, 2.5, 4, 5.5, 7 and 8.5 pixel radii with 8, 16, 24, 32, 40 and 48 samples respectively. The samples were extracted using bilinear sampling. The MSLBF classifiers for this task comprised 276 binary classifiers. The results here correspond to the application of the four classifiers to a separate testing set comprising the images not used in training.

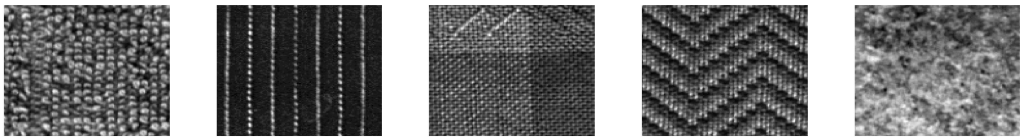


Figure 5.4: Examples from the Outex_00000 texture suite.

Table 5.1 presents the classification results in percentages for the four classifiers for each of the texture classes. The MSLBF models were applied with different numbers of features selected between 1 and 8. The MSLBF results are presented along with a value in brackets indicating the number of features required for all classifiers before the corresponding percentage score was reached for that class. Although this table suggests that different classes need only their own number of features, in practice the upper limit was necessary across all classes to obtain a stable performance (i.e. MSLBF+BHIM required 4 bits for "carpet009" meaning 4 bits were retained for all classes). This is likely an issue with the classification procedure which does not affect the utility of the table in demonstrating how very few bits were generally required per class for strong discriminative power. The minimum number of features required (between 1 and 8) for the best score obtained for each class was recorded. Table 5.2 provides the overall best scores across all classes along with the number of features required to achieve those scores, averaged over all classes. It can be seen that all three MSLBF classifiers outperformed the vanilla combined-predicate LBP. Only the MSLBF+BHIM combination achieved a perfect score and with a lower number of average features required per class. The highest number of features required for BHIM was 4 (class "carpet009"), 5 for CMIM and 6 for AdaBoost. Consequently the MSLBF+BHIM combination constituted a more compact and effective discriminative model than the other combinations. To compare the relative strength of the four classifiers, average histogram distances for each class was computed. Figure 5.5 plots discriminative strengths of BHIM, CMIM and AdaBoost which shows consistently superior model separation for BHIM

generated models.

Class	*LBP* (%)	*MSLBF* +BHIM	*MSLBF* +CMIM	*MSLBF* +AdaBoost
canvas001	100	100 (1)	100 (1)	100 (1)
canvas002	100	100 (1)	100 (1)	100 (1)
canvas003	100	100 (1)	100 (1)	100 (1)
canvas005	100	100 (2)	100 (2)	100 (2)
canvas006	100	100 (1)	100 (1)	100 (1)
canvas009	100	100 (1)	100 (1)	100 (1)
canvas011	100	100 (2)	100 (2)	100 (2)
canvas021	100	100 (2)	100 (2)	100 (2)
canvas022	100	100 (2)	100 (2)	100 (2)
canvas023	80	100 (2)	100 (2)	100 (3)
canvas025	100	100 (1)	100 (1)	100 (1)
canvas026	100	100 (1)	100 (1)	100 (1)
canvas031	100	100 (2)	100 (2)	100 (2)
canvas032	100	100 (2)	100 (2)	100 (2)
canvas033	80	100 (2)	100 (3)	100 (2)
canvas035	60	100 (1)	100 (1)	100 (1)
canvas038	100	100 (2)	100 (3)	100 (3)
canvas039	100	100 (1)	100 (1)	100 (1)
tile005	70	100 (2)	100 (5)	100 (6)
tile006	100	100 (2)	100 (4)	100 (6)
carpet002	100	100 (1)	100 (1)	100 (1)
carpet004	100	100 (1)	100 (1)	100 (1)
carpet005	100	100 (2)	100 (2)	100 (2)
carpet009	100	100 (4)	90 (5)	90 (3)

Table 5.1: Classification of Outex 00000 database which contains variations of canvas, tile and carpet type textures. The multipredicate $LBP^{riu2}_{8,1} + ^{riu2}_{16,3} + ^{riu2}_{24,5}$ classifier is generated from training data comprising predicates at 1, 3 and 5 pixels radius with 8, 16 and 24 samples per predicate. The MSLBF classifiers were generated with three different feature selection algorithms on a larger training set comprising six predicates at 1, 2.5, 4, 5.5, 7 and 8.5 pixels radius with 8, 16, 24, 32, 40 and 48 samples respectively. The MSLBF results are given along with a value in brackets indicating the lowest number of features required to achieve the corresponding success rate for that class (up to a maximum of 8).

Classifier	Success (%)	Mean no. features
$LBP^{riu2}_{8,1}+^{riu2}_{16,3}+^{riu2}_{24,5}$	95.4	-
MSLBF+BHIM	100	1.625
MSLBF+CMIM	99.6	1.958
MSLBF+AdaBoost	99.6	2

Table 5.2: Overall success rate of the four classifiers with Outex_00000 along with the average number of features needed to gain the best scores shown in Table 5.1 per class. The LBP classifier is constructed from smaller sample areas than MSLBF.

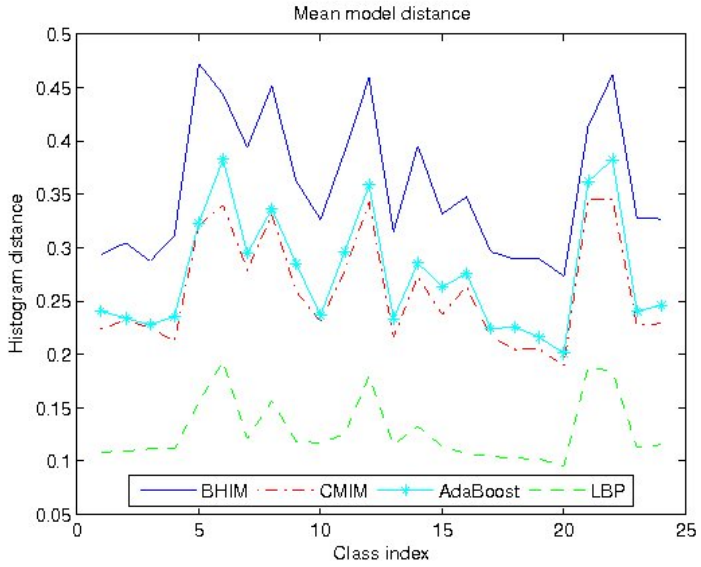


Figure 5.5: Average histogram separation per class for LBP and MSLBF models generated by BHIM, CMIM and AdaBoost. The mean histogram distance for each class for Outex_00000 is plotted. MSLBF+BHIM has significantly larger between-class distances.

5.3.2 Face recognition

A more challenging problem of face recognition given face images captured under large variations in lighting and 3D pose was also considered. The ORL face database [186] was employed for comparing the MSLBF approach to standard multipredicate $LBP^{u2}_{8,1}+^{u2}_{16,3}+^{u2}_{24,5}$. This is a relatively small database comprising 400 unregistered images of 40 people with 10 samples for each person. The samples are greyscale and sized at 92×112 pixels. They contain large within-class variance in lighting, pose and appearance due to the presence/absence of glasses/facial hair and different times of capture (see Figure 5.6).

The data was split up with five samples per person used for training (even indices) and the other five used for testing (odd indices). LBP has been previously applied using a windowed

Figure 5.6: Examples from the ORL face database demonstrating within-class variations of appearance, lighting and/or pose.

approach (Ahonen et al. [2]) to model different facial regions separately with good results. Although modelling different facial regions separately and weighting them according to importance was shown to demonstrate better classification (Shan et al. [193]), here the faces were modelled globally to make the problem more generic (independent of ad hoc region segmentation) and to gauge the benefit of the pairwise-coupled MSLBF approach. The same sample sizes were used for both an LBP and three MSLBF classifiers, again trained with BHIM, CMIM and AdaBoost for up to 8 features, with predicates in the training data being formed at 1, 3 and 5 pixel radii with 8, 16 and 24 samples respectively. Table 5.4 shows results for applying an $LBP^{u2}_{8,1}+^{u2}_{16,3}+^{u2}_{24,5}$ classifier along with the three MSLBF combinations to the database. The MSLBF models were increased in the number of features selected and the percentage values are accompanied by a bracketed number indicating the number of bits required to obtain the corresponding percentage for that class. Table 5.3 shows the overall results averaged over classes. The MSLBF combinations outperformed the vanilla LBP classifier with the MSLBF+BHIM combination proving the best. The MSLBF+AdaBoost combination came second with the same number of average bits per class but inferior classification success. MSLBF+CMIM did not improve on LBP by any significant margin.

Figure 5.7 demonstrates the mean histogram distances for each face class for LBP and the three MSLBF classifiers as combined with BHIM, CMIM and AdaBoost. As with textures, MSLBF+BHIM shows better histogram distances.

Classifier	Success (%)	Average bits
$LBP^{u2}_{8,1}+^{u2}_{16,3}+^{u2}_{24,5}$	87	-
MSLBF+BHIM	93	3.9
MSLBF+CMIM	87.5	4.2
MSLBF+AdaBoost	91	3.9

Table 5.3: Overall success rate of the four classifiers with the ORL database along with the average number of bits needed to gain the best scores. The MSLBF classifiers were constructed from the same sample regions as the LBP classifier.

Class	*LBP* (%)	*MSLBF* +BHIM	*MSLBF* +CMIM	*MSLBF* +AdaBoost
1	40	100 (6)	40 (8)	80 (5)
2	100	100 (2)	100 (4)	100 (3)
3	100	100 (7)	80 (5)	100 (8)
4	100	80 (5)	80 (8)	80 (7)
5	100	100 (4)	100 (5)	100 (6)
6	100	100 (3)	100 (7)	100 (4)
7	100	100 (1)	100 (1)	100 (1)
8	100	100 (3)	100 (6)	100 (3)
9	100	100 (5)	100 (8)	100 (6)
10	100	100 (5)	100 (7)	100 (5)
11	100	100 (1)	100 (1)	100 (1)
12	100	100 (5)	100 (6)	100 (5)
13	80	100 (3)	100 (4)	100 (4)
14	100	100 (2)	100 (2)	100 (2)
15	40	100 (7)	100 (2)	100 (2)
16	80	80 (6)	60 (7)	80 (7)
17	80	100 (2)	100 (4)	100 (5)
18	100	100 (5)	80 (4)	100 (4)
19	100	100 (3)	100 (2)	100 (3)
20	100	100 (3)	100 (4)	100 (4)
21	80	100 (2)	100 (6)	100 (2)
22	100	100 (3)	100 (4)	100 (2)
23	60	100 (7)	100 (6)	100 (5)
24	100	100 (6)	80 (1)	100 (6)
25	100	100 (4)	100 (4)	100 (6)
26	80	100 (2)	100 (2)	100 (2)
27	100	100 (2)	100 (2)	100 (2)
28	100	100 (8)	100 (8)	100 (7)
29	100	100 (2)	100 (5)	100 (4)
30	100	100 (6)	100 (4)	100 (5)
31	40	60 (8)	40 (5)	40 (4)
32	100	100 (2)	100 (2)	100 (2)
33	100	100 (3)	100 (7)	100 (6)
34	100	80 (6)	60 (2)	80 (4)
35	60	100 (2)	80 (1)	80 (1)
36	100	80 (7)	60 (2)	60 (2)
37	20	20 (2)	20 (2)	20 (2)
38	100	100 (3)	100 (5)	100 (3)
39	100	100 (3)	100 (4)	100 (5)
40	20	20 (1)	20 (1)	20 (1)

Table 5.4: Classification results for the ORL face database. An $LBP^{u2}_{8,1} +^{u2}_{16,3} +^{u2}_{24,5}$ classifier is generated from training data comprising predicates at 1, 3 and 5 pixels radius with 8, 16 and 24 samples per predicate. The MSLBF classifiers were generated with three different feature selection algorithms on the same data set. The numbers in brackets indicate the lowest number of features required to achieve the corresponding success rate for that class (up to a maximum of 8).

5.3.3 Comparison of feature selection methods

Figure 5.7 showed the MSLBF+AdaBoost combination to be close to MSLBF+BHIM. In order to further examine the effectiveness of feature selection with these combinations, additional experiments (with ground truth) were designed to compare an efficient implementation of BHIM with CMIM and AdaBoost on binary feature selection tasks. CMIM (Conditional Mutual Infor-

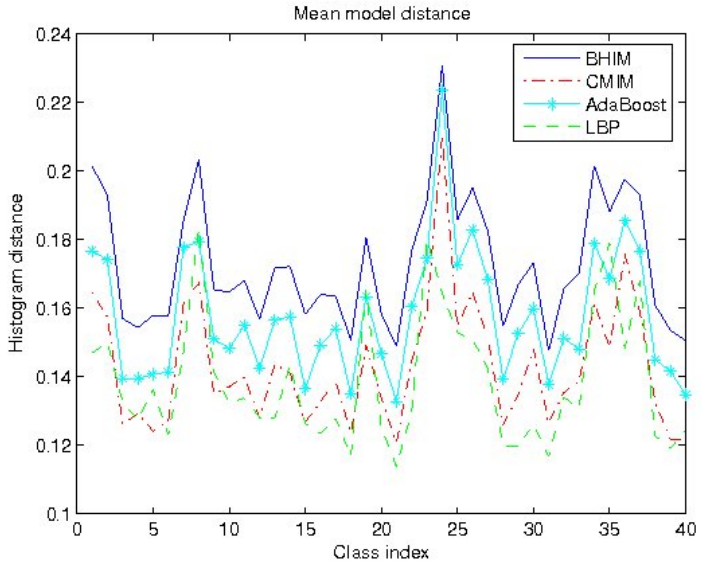


Figure 5.7: Average histogram separation for LBP and MSLBF models for the ORL face database. BHIM, CMIM and AdaBoost were compared for selecting features for MSLBF models. As with textures, the BHIM features show larger histogram distances.

mation Maximisation) [55] is a filter that employs information theory in a rigorous manner to select features correlated with class labels with minimal redundancy amongst themselves. Ideally, the best set of K features $\{b_1, b_2, ..., b_K\}$ given training data are those that minimise the conditional entropy $H(Y|X_{b_1}, X_{b_2}, ..., X_{b_K})$ where Y is the random variable corresponding to class labels. The experiments here made use of a Matlab implementation of the fast CMIM algorithm. Also implemented for testing was a very efficient binary AdaBoost algorithm based on [61].

AdaBoost can be used as a feature selection algorithm by considering all features in a feature pool to be a weak learner. Incorrectly classified samples for a selected feature (weak learner) are taken to be those that hold the least frequent value given the histogram formed from the whole data set for that feature. The best feature is the one with the maximum binary histogram separation between the two data sets (with binary histograms formed from the weighted samples).

For the objective of this experiment, synthetic datasets were created with each binary feature drawn from a flat distribution. 12 features were randomly selected and a 2^{12}-bin histogram randomly generated for the joint distribution for each class. The distributions for each class were overlapped to random degrees so that a set of joint values had a positive probability for both classes. Samples were drawn from these distributions and the corresponding 12-bit binary strings placed into the data at the selected positions as samples. These embedded structures provided

a target set of strong features among random ones for algorithms to find. Three factors were examined; (1) the "quality" of features selected, measured by the conditional entropy of the class variable given the selected features, $H(Y|b_1, b_2, ..., b_K)$, (2) the percentage of features selected that matched the features randomly embedded and (3) computation time for selection. Each of these three factors were plotted against: (a) feature pool size varying between 100 and 800, (b) number of training samples for each class ranging between 10000 and 80000 and (c) number of features an algorithm was required to select from 2 to 12. Each parameter configuration was applied to 100 randomly generated pairs of distributions with random structure and the results averaged. Default parameters were 100 for feature pool size, 10000 for number of samples for each class, 12 features embedded and 12 features to select. Parameters that were not being adjusted took these default values.

Figures 5.8, 5.9 and 5.10 show the results. From Figure 5.8, it can be seen that BHIM outperformed both CMIM and AdaBoost across varying feature pool sizes. Plot (a) of the figure shows that the average remaining entropy for BHIM-selected features correlated closely with the remaining entropy given the embedded features and Plot (b) that the features selected were strongly (and often perfectly) correlated with the features that were randomly embedded. CMIM and AdaBoost were comparable but selected significantly lower quality features compared to BHIM. Plot (c) shows that in terms of computation time, all three algorithms are linear in the size of the feature pool.

Figure 5.9 plots performance against varying training-set sizes. The figure demonstrates the same trends as with varying feature pools in Figure 5.8. Plot (a) of the figure shows that again the average remaining entropy given BHIM-selected features correlated closely with the remaining entropy given the embedded features. Plot (b) demonstrates that selected features are close to perfectly correlated with those embedded. Plot (c) shows linearity in computation time with increasing number of samples in the training sets.

Figure 5.10 plots performance against increasing numbers of features to select. While entropies and embedded feature selection showed similar trends for all three algorithms as for varying feature pools and training-set sizes in Figures 5.8 and 5.9 respectively, BHIM showed a weakness in its exponential increase in computation time with the number of features to select. This is because the main loop in BHIM involves comparing features against all values shared between two data sets given the previously selected features. The maximum possible number of

shared values is equal to 2^k where k is the number of previously selected features. Consequently, each successive feature selection cycle can potentially double the number of cycles in the main loop up to the size of the training set. However, the limited size of training sets restricts the number of features that can be considered reliably when calculating expected histogram distances, resulting eventually in a linearisation of computation time.

5.4 Discussion

For the texture and face recognition experiments, each input image was applied to each of the pairwise-coupled classifiers with the highest histogram intersection value being added to the score of the corresponding class. Alternative scoring procedures could have been employed. We also tested two others, namely (for each pairwise classifier) (1) adding a 1 to the score of the highest match and (2) adding the histogram intersection values to both classes. However, in practice there were no distinct differences between them, with the portrayed scoring method resulting in slightly better performance than the other two. We also tested the use of Kullback-Leibler divergence as a method for histogram matching, both for computing expectations for feature selection in Equation 5.7 and the pairwise-coupled matching of input images. Again, there were no clearly notable differences. Consequently, the use of histogram intersection as a far cheaper method were retained.

In general for practical classification tasks, the pairwise-coupled approach for an N-class problem results in $\frac{1}{2}N(N-1)$ steps that consist of classifier-forming and matching. As such, the number of classifiers increases quadratically with the number of classes, although the computational time for classifying an input is linear with the number of classifiers. However, with an unoptimised Matlab implementation on a very modest single-core desktop computer, classification of an input image for a 24-class texture experiment involving 276 binary classifiers (see Section 5.3.1) required only a fraction of a second. Classification of an input for a forty-class face recognition task involving 780 classifiers (Section 5.3.2) required just over a second on average per input. The low number of features for each classifier helps to keep computation time down both in terms of histogram assembly and comparison.

There are two main drawbacks to the pairwise-coupled approach. Firstly, in the experiments, stable results required the same number of features to be used for all classes despite the varying numbers of features required for a given error per class. Secondly, the complete separation be-

tween the training of individual binary classifiers does not preclude the possibility of histograms for two classes being similar despite being constructed from completely different features. This can lead to a degradation in performance and effectively dilute the potency of the features originally selected. This problem suggests that an extra element is required to further enhance the contextual setting for a classification task, such as a preliminary global step to narrow down the possibilities. As such, further investigation to explore alternative classification methods, such as tree-based classifiers or a one-to-many approach, would be useful.

At each iteration the BHIM algorithm finds features whose class-conditional binary histograms are maximally divergent (have minimal histogram intersection) *when previously selected features fail to discriminate*. In that regard, it shares some superficial similarities with AdaBoost in that "difficult" samples which cannot be correctly discriminated by the current set of chosen features are isolated and focused on in the next iteration of the algorithm. It is important to emphasise that at each iteration, *all* of the current chosen features are considered jointly in isolating difficult samples, which invokes a greater emphasis on lowering redundancy than the CMIM approximation technique.

As shown in Section 5.3.3, the BHIM algorithm is linear in computation time with the size of the feature pool or the size of the training set. However, it is essentially exponential-time in the number of features to select, which is generally an undesirable quality. This is due to the potential doubling (in the worse case) of terms employed in computing the expectation step for each feature candidate (Equation 5.7) However, this is tempered by two factors. Firstly, the presence of strong features in the training set can significantly reduce the number of terms required in the expectation computations since the overlap between the joint distributions is more strongly reduced by each feature selected. Secondly, the finite size of training sets places a natural limit on the number of features that may be jointly considered for reliable estimates of class-conditional binary histograms. This is because, in the worst case, for two training sets of M samples each there are a maximum of M possible shared values regardless of the number of features selected. As the number of features increases, the number of samples per shared value for estimating histogram distances reduces, eventually resulting in unreliable estimates. Consequently, the "window" of the expectation calculation w, involving only at most the w previously selected features, may be estimated as $w = log_2 \frac{M}{v}$, where v is the desired minimum average number of samples per shared value and M is the number of samples in the training set for a class. Consequently, for a binary

histogram and assuming ten samples per bin for a representative sample, v may be set to 20. Employing this approach has the twin benefits of (a) ensuring that class-conditional histograms are estimated from adequate numbers of data and so preventing overfitting and (b) capping computational time such that an initial exponential increase eventually becomes *linear*.

While BHIM in the tests conducted here demonstrated great strength both with random and real experimental data, the improvement in model separation shown does not appear to translate as strongly to classification performance in overall MSLBF terms (as in Figure 5.7). However, the results imply that improving the quality of feature pools would automatically draw on the strengths of the BHIM algorithm as opposed to the others. Although it appears to show its potency in extensive experimentation, a further theoretical exploration of the limitations of this algorithm would be important for an objective understanding of its strengths.

5.5 Summary

In this chapter, two contributions were described:

1. A new LBP-type model was introduced known as Multiscale Selected Local Binary Features (MSLBF);

2. A novel binary feature selection algorithm was described known as Binary Histogram Intersection Minimisation (BHIM).

MSLBF models are compact LBP-type predicates that capture the joint statistics of selected textons across scales. They are generated through the use of feature selection to select strongly discriminative textons from a pool collected from arbitrary neighbourhoods. A pairwise-coupling classification approach is taken to enable greater specificity in selected features and simplify the feature selection process. The benefits of this method are:

1. It may be viewed as a natural and intuitively appealing context-dependent sampling strategy which focuses on image structures that are most relevant depending on the comparison being made;

2. Selecting individual pixel features rather than taking combined spatially contiguous groups of features with possible redundancy enables more compact and descriptive models;

3. Circular feature pools at any scale and angular resolution can be incorporated into the training data.

The BHIM algorithm is a greedy feed-forward filter method designed specifically for two-class binary problems and ideally suited to generating MSLBF classifiers. It has several advantages over previous feature selection algorithms used for this kind of task:

1. Unlike other binary feature selection algorithms, it selects new features on the basis of their predictive power given *all* of the previously selected features;

2. It selects a typically small number of features with strong discriminative power and minimal redundancy in an information-theoretical sense;

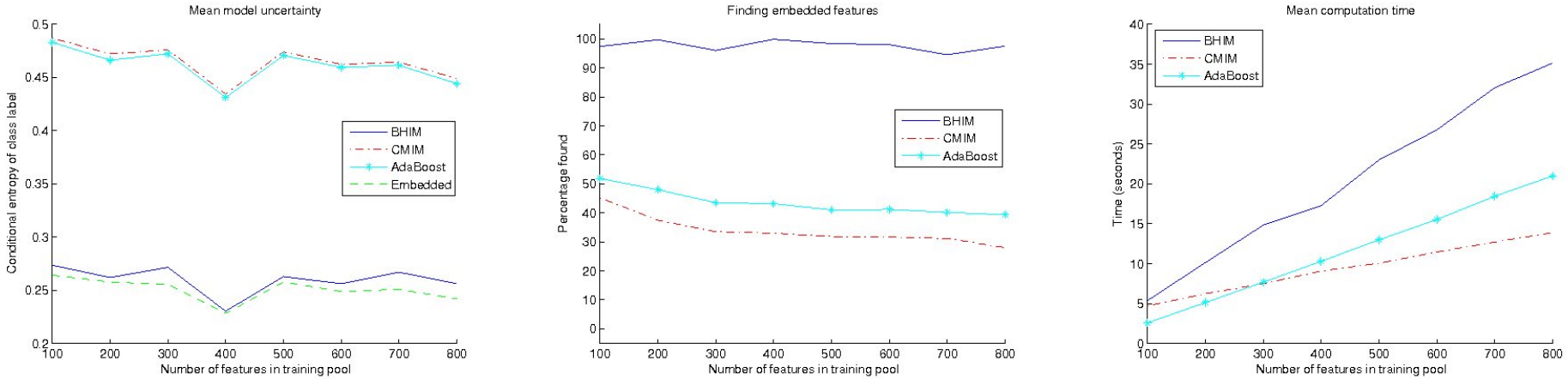
3. It does not require expensive entropy-type computations;

4. It is relatively computationally inexpensive and expends fewer resources when stronger features are available;

5. Is linear in computation time in the long term due to the natural practicalities of limited training data.

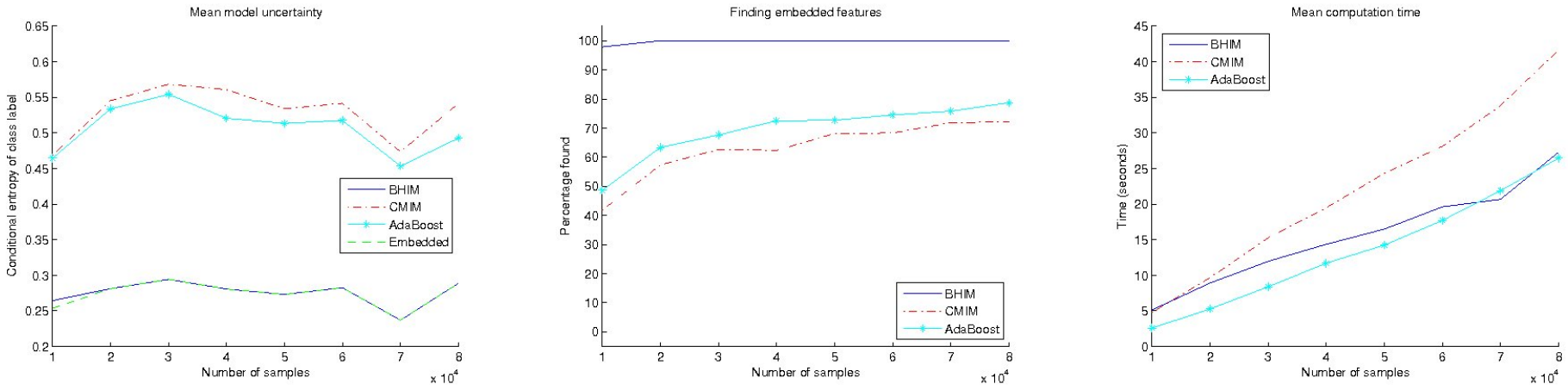
We conducted three experiments, two on real-world classification tasks for comparing LBP and MSLBF classifiers generated with different feature selection algorithms. The third experiment was designed to test the effectiveness of BHIM as compared with AdaBoost and CMIM on synthetic data containing pre-selected embedded features. The first classification experiment was conducted for textures, where it was shown that the MSLBF approach employing BHIM for feature selection enabled perfect classification with relatively few selected features. The second classification experiment was conducted for faces on a fairly difficult face database that was more challenging for the traditional LBP classifier. This showed improvement for MSLBF based classifiers with the best result provided by the MSLBF classifier generated from BHIM selected features. The final experiment was conducted on random synthetic datasets with randomly generated distributions embedded at 12 random positions in the feature pool. The BHIM algorithm clearly outperformed both AdaBoost and CMIM in finding the strongest features. To conclude, in these experiments the BHIM algorithm consistently demonstrated its ability to select stronger features than either CMIM or AdaBoost in terms of a concrete information theoretical measure. The algorithm expends variable computational resources depending on the strength of the data

available (stronger features require less computation to find) and has limits on the exponential nature of reliable expectation estimates at each step, enabling linearity of computation time in the long term. A major reason for the overall efficiency of the algorithm is that the computationally cheap two-bin histogram intersection is all that is required and actual entropy computations are unnecessary.



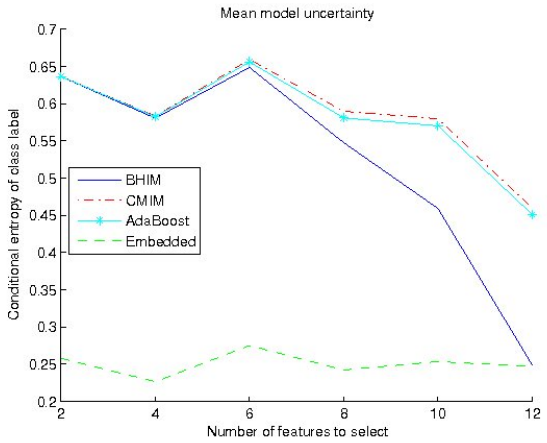
(a) (b) (c)

Figure 5.8: Comparison of BHIM, CMIM and AdaBoost plotted over varying feature pool sizes and averaged over 100 random two-class datasets containing 12 embedded features with each class represented by 10000 samples. The plots denote; (a) average remaining entropy of the class variable given the selected features, (b) average proportion of selected features that match the embedded features and (c) the time required to select 12 features. BHIM was clearly more successful at finding the embedded features as shown in plots (a) and (b), without being influenced by large numbers of distractors in the pool. Computation time was linear with the size of the feature pool.

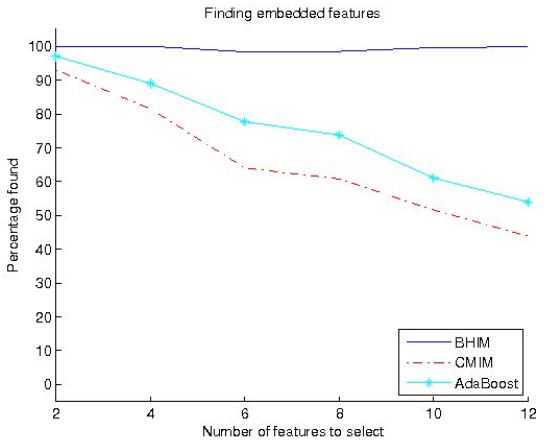


(a) (b) (c)

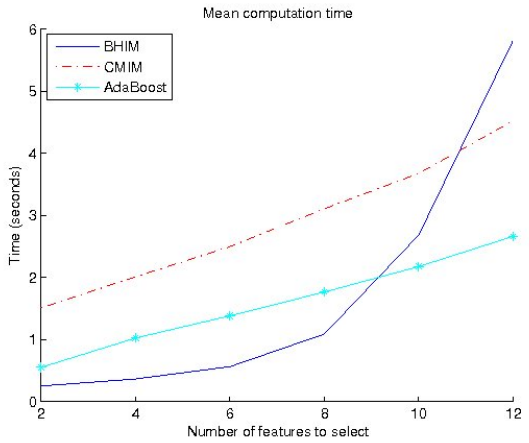
Figure 5.9: Comparison of BHIM, CMIM and AdaBoost plotted over varying numbers of training samples per class and averaged over 100 random two-class datasets with 12 embedded features. The plots denote; (a) average remaining entropy of the class variable given the selected features, (b) average proportion of selected features that match the embedded features and (c) the time required to select feature subsets. As for Figure 5.8, BHIM was more successful at finding the embedded features without being affected by the number of training samples. Computation time was linear with the size of the training set.



(a)



(b)



(c)

Figure 5.10: Comparison of BHIM, CMIM and AdaBoost plotted over varying numbers of features to select and averaged over 100 random two-class datasets containing 12 embedded features with each class represented by 10000 samples. The plots denote; (a) average remaining entropy of the class variable given the selected features, (b) average proportion of selected features that match the embedded features and (c) the time required to select feature subsets. Here, it can be seen in (a) that as more and more features were recovered by BHIM, the closer the remaining entropy dropped to that of the embedded set. This is to be expected since the reference entropy was computed from all 12 embedded features. Plot (b) illustrates that the features selected by BHIM were always strongly correlated with those embedded. Computation time for BHIM was exponential with the number of features to select as opposed to linear for CMIM and AdaBoost. However, total time taken to recover all 12 features was not excessively higher.

Chapter 6

Conclusions

Visual sampling has long been recognised as an intrinsic component of human visual experience, directing focus of attention in a dynamic and controlled way in order to balance various factors such as efficiency, information gain and contextual relevance. These concepts are also inherently applicable to the computer vision domain, where visual *sampling strategies* may be considered as the foundation for object association tasks such as tracking and recognition. These strategies may be viewed as *filtering* processes. This thesis focused on the development of adaptive visual sampling strategies for both tracking and recognition and in the process built upon previous methods in both of these areas to improve intuitive perspective as well as improve performance.

6.1 Colour feature sampling for tracking

Tracking tasks can be addressed by dynamic sampling strategies that generally comprise two filtering steps at each point in time: (a) focusing of attention through predictive mechanisms to reduce the number of samples (pixels) under consideration; and (b) filtering of remaining samples to evaluate their association with a model and produce final estimates of object states such as position and size. The reliability of state estimates are dependent not only on the predictive step but also on the quality of the measurements, which in turn are rooted in the relevance of the model used for sample filtering. Many techniques do not adequately address the need for such models to adapt to real-world conditions, where relevance is a fluid property liable to rapid and frequent change.

We addressed this by developing a framework for the adaptive statistical modelling of object

features; specifically, colour values in Hue-Saturation space. These models were used for real-time and robust tracking of multi-colour objects under changing lighting conditions. Colour features have been used for a variety of tasks such as segmentation [201], tracking [138] and object recognition [134, 212]. We described previous colour-based methods appropriate to this area and addressed two specific limitations:

1. The use of non-parametric models which can be sensitive to small training sets;
2. The lack of a dynamic sampling strategy which prevents object models from maintaining relevance under changing conditions over time.

In doing so we developed a sampling strategy framework for tracking based on pixel feature statistics and applied it for colour. This involved five components:

1. The use of semi-parametric Gaussian mixture models to capture multi-colour distributions. This has the benefit of enabling a more flexible selection of model order than histogram bin sizes as well as being better able to deal with the small quantities of data found during tracking;
2. An algorithm known as *Iterative Model Order Selection* (IMOS) for automatically selecting the number of components of a Gaussian mixture model. This employs a cross-validation approach to incrementally add components and terminate when the model begins to overfit;
3. A fast filtering-based sampling strategy that employs the model to sample from the image of a sequence and estimate object position and size. This operates in real-time on extremely modest hardware;
4. A Bayesian formulation for context-dependent pixel classification that more rigorously employs adaptive models of both object and background. This improves the accuracy of pixel filtering and consequently estimates of object state;
5. A mechanism to facilitate a dynamic sampling strategy by automatically adapting a mixture model to deal with colour changes caused by changing lighting conditions. This also includes a more intelligent mechanism for detecting tracking failure and suspending adaptation in order to prevent model drift.

We demonstrated the effectiveness of adaptive over static models on sequences depicting the tracking of faces against changing backgrounds under dynamic lighting conditions and undergoing partial occlusion with selective adaptation. We also demonstrated the Bayesian filtering formulation by tracking a multi-coloured torso with segmentation to illustrate the accuracy of pixel classification.

6.1.1 Future work

The colour models used here do not capture spatial structure, which has the advantage of enabling greater flexibility for tracking under geometric transformations of both object and scene. The drawback is that geometrical structural differences between object and distractors cannot be exploited for discrimination. Consequently, the use of geometric features in conjunction with photometric is likely to be valuable. Furthermore, such different cues are likely to need controlled mediation in order to obtain a satisfactory result since different cues will exhibit different levels of relevance at different times. Although we began to explore mechanisms for this in Chapter 4 using only photometric features, it seems likely that the use of geometry-invariant features such as SIFT [121] are crucial to the effort. The framework may then be extended by using colour pixel filtering results to filter geometric features.

The Iterative Model Order Selection algorithm is a simple and intuitively appealing approach to selecting model order. However, given the dynamic nature of tracking and acknowledged by the use of adaptive models, it is clear that, apart from the actual distributions over the feature space, optimal model order itself is likely to change over time. For example, the number of distinct colours of an object, each of which produces a peak in the corresponding distribution in feature space, may change with object pose. Consequently, it is desirable to further improve adaptation methodology to cater for this requirement. This would necessarily involve procedures for not only adding components but also removing them when appropriate.

6.2 Multi-feature sampling for tracking

Tracking in a cluttered scene under unstable lighting is hindered by the intrinsic instability and transience of features as a useful discriminator for the target object. Furthermore, the robustness of different types of features is highly context-dependent, with photometric and geometric environmental conditions inducing different complications. For example, colour is sensitive to

changes in lighting whilst shape and texture may be drastically altered during pose transitions. Additionally, in cluttered scenes dynamic distractors can significantly affect the relevance of specific features over time, e.g. colour may perform adequately when a red target object is tracked against a non-red background but shape may be more discriminative when the target moves into a red-coloured area. Consequently, in the absence of truly robust features, a successful tracker will not only take measures to alleviate the problems of model-drift but also to utilise the features most likely to be useful at any given time.

We extended the sampling strategy for tracking described in Chapter 3 by incorporating a *third* filtering step to address the issue of selecting appropriate features during tracking. Whilst previously filtering was performed purely in the spatial domain, here the notion was extended to another dimension, the *featural* domain of the Spatial-Featural Volume (SFV) as illustrated in Figure 1.8. This extra dimension constituted an extra source of pixel samples, with each spatial image coordinate now indexing a pool of values derived from various transformations of the raw image colours. Previous work had addressed the issue of selection from this pool for tracking (e.g. [29, 4, 115, 72]) with the following limitations:

1. In choosing the most relevant features in each frame, features are ranked using metrics or boosting methods which do not address issues of redundancy amongst those selected (see Section 2.1.2);

2. Model drift is inadequately addressed by using static reference models based on unrealistic assumptions of long-term relevance.

Here we addressed these problems within a framework called *Adaptive Multi-Feature Association* (AMA) consisting of two components:

1. A more reliable feature ranking method called *Attribute-based Feature Ranking* (AFR) which consists of a combination of two computed attributes per feature to reduce redundancy;

2. A mechanism called *Multiple Selectively-adaptive Feature Models* (MSFM) for maintaining multiple longer-term reference models that are selectively adapted on-line to avoid model drift.

We demonstrated the effectiveness of this framework in challenging tracking scenarios depicting small target objects in cluttered environments undergoing severe lighting changes and

occlusions and showed that the features selected tended to be more descriptive and the use of adaptive reference models with dynamic prior re-evaluation was a more effective approach to balancing short and long-term evidence to maintain relevance for the sampling strategy.

6.2.1 Future work

We employ canonical correlation analysis (CCA) as an approximate method for estimating correlation between features through a comparison of samples. This has the advantage that distributions need not be estimated, avoiding the need for large amounts of data which are typically unavailable. Furthermore, it is computationally much cheaper than most feature selection algorithms, an important factor in a time-critical application such as object tracking. However, despite the experimental results given here, a further objective study of the "quality" of features recovered using our CCA-based attribute method would be desirable as well as the effects of modifying the weights for each of the attributes in the ranking process.

Our framework includes the dynamic re-evaluation of the priors used for pixel classification (Equations 4.13 and 4.14) and updating of feature reference models (Equation 4.6). This is currently based on comparing the data for foreground and background across frames and the relative overlap between them as indicative of tracker errors or object occlusion. Additionally, the mechanism employed for model adaptation served to illustrate the concept we proposed in this chapter despite being rather simple. Further investigation should explore the limits of this approach and more sophisticated methods developed for controlling adaptation, including alternative mechanisms for distinguishing different causes of tracker failure and their corresponding implications for reference model updates.

In our experiments we employed a rather limited set of features, all of which are photometric. We employed these for simplicity and to demonstrate our framework. It would be interesting to conduct further experiments comprising larger feature pools incorporating geometric statistical features such as oriented gradients [41] as well as alternative colour spaces. Further, the integration of powerful invariant features such as SIFT [121] into the overall framework might provide a strong boost to performance. This would require more research to explore the best ways for features not yielding statistical distributions to be used as part of the overall sampling strategy.

Currently, this framework focuses on the tracking of a single object. A natural way forward is to integrate data association techniques such as Joint Probabilistic Data Association Filters (JPDAFs) [60] to handle multiple tracked objects simultaneously in difficult scenarios and in a

unified and rigorous manner.

6.3 Local Binary Pattern feature sampling for recognition

Visual recognition, like visual tracking, finds its basis in the problem of object association. Although tracking is usually concerned with distinguishing between the foreground object being tracked and the background, this can naturally extend to discriminating between different foreground objects under difficult conditions, as addressed by Song et al. [204] who employ object classification to perform disambiguation when tracking people in crowded scenes. In either case, there is a fundamental requirement for a sampling strategy that scrutinises the most appropriate parts of an object in order to both reliably and efficiently perform discrimination.

We addressed some of the limitations of one specific type of methodology for object discrimination, namely Local Binary Pattern (LBP) based recognition applications. LBP methods have been used extensively for a huge range of applications, including texture discrimination [130, 155], texture segmentation [172] and recognition of facial identity [2] and expression [53, 193]. We described the limitations of previous LBP methodologies and argued that they are borne from a fundamentally inflexible approach to modelling LBP statistics which:

1. Either limit the spatial area over which models may capture information or otherwise average over potentially useful details;
2. Incorporate possibly redundant information which wastes resources;
3. Decouples statistics in an ad hoc manner;
4. Builds models in a spatially non-selective, context-agnostic way which further impacts on classification accuracy and betrays the inherent importance of adaptive visual sampling approaches to discrimination.

We then proposed a more natural framework which solves all of these problems and may be added without modification to many existing LBP-type methods which involve modelling distributions of jointly encoded binary sequences such as [127, 118]. This framework involves:

1. A novel feature selection algorithm designed for binary data, called *Binary Histogram Intersection Minimisation* (BHIM), which is capable of finding stronger, less-redundant feature subsets than two state-of-the-art algorithms for binary feature selection;

2. The encoding of selected features to form distributions from context-dependent spatial topologies called *Multiscale Selected Local Binary Features* (MSLBF);

3. the use of MSLBF models in a pairwise-coupling [82] scheme to enable the most appropriate samples to be used depending on the two classes being compared.

We demonstrated the effectiveness of the framework over traditional LBP approaches in two specific recognition applications; namely texture classification and face recognition. Simultaneously, we showed the improved descriptiveness of the feature selected by the BHIM algorithm over two established algorithms used for binary feature selection. We also presented a third experiment that performed an extensive comparison of the three feature selection methods on synthetic data and showed the improved performance of BHIM without an excessive increase in computational cost.

6.3.1 Future work

There are two main drawbacks to the pairwise-coupled approach. Firstly, in experiments, stable results required the same number of features to be used for all classes despite the varying numbers of features required for a given error per class. Secondly, the complete separation between the training of individual binary classifiers does not preclude the possibility of histograms for two classes being similar despite being constructed from completely different features. This can lead to a degradation in performance and effectively dilute the potency of the features originally selected. This problem suggests that an extra element is required to further enhance the contextual setting for a classification task, such as a preliminary global step to narrow down the possibilities. As such, further investigation to explore alternative classification methods, such as tree-based classifiers or a one-to-many approach, would be useful.

While the BHIM algorithm demonstrated great strength both with random and real experimental data, the improvement in model separation shown did not appear to translate as strongly to classification performance in overall MSLBF terms. However, the results imply that improving the quality of feature pools would automatically draw on the strengths of the BHIM algorithm as opposed to the others. Although it appears to show its potency in extensive experimentation, a further theoretical exploration of the limitations of this algorithm would be important for an objective understanding of its strengths.

Appendix A

On-line Gaussian Mixture Adaptation

Here we derive Equations 3.15 and 3.16 in Section 3.6.1 for adapting Gaussian mixtures on-line. At time t we desire a mixture generated from the data from the $L+1$ frames up to and including frame t. This can be computed as a weighted average of the data from those frames. Consequently, the parameters for component j can be formulated like so:

$$\boldsymbol{\mu}_t = \frac{\sum_{\tau=t-L}^{t} \psi^{(\tau)} \left\{ \frac{1}{\psi^{(\tau)}} \sum_{\mathbf{x} \in X^{(\tau)}} p(j|\mathbf{x})\mathbf{x} \right\}}{\sum_{\tau=t-L}^{t} \psi^{(\tau)}} \tag{A.1}$$

$$\boldsymbol{\Sigma}_t = \frac{\sum_{\tau=t-L}^{t} \psi^{(\tau)} \left\{ \frac{1}{\psi^{(\tau)}} \sum_{\mathbf{x} \in X^{(\tau)}} p(j|\mathbf{x})(\mathbf{x}-\boldsymbol{\mu}_{t-1})^T(\mathbf{x}-\boldsymbol{\mu}_{t-1}) \right\}}{\sum_{\tau=t-L}^{t} \psi^{(\tau)}} \tag{A.2}$$

where $\psi^{(\tau)}$ denotes the sum of the posterior probabilities of the data $X^{(\tau)}$ in frame τ:

$$\psi^{(\tau)} = \sum_{\mathbf{x} \in X^{(\tau)}} p(j|\mathbf{x})$$

Denote D_t as the sum of the sum of the posterior probabilities ψ^{τ} over the $L+1$ frames up to and including frame t:

$$D_t = \sum_{\tau=t-L}^{t} \psi^{(\tau)}$$

We wish to derive recursive expressions for the component parameters $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$. Observing that Equations A.1 and A.2 both take the form of:

$$\boldsymbol{\theta}_t = \frac{1}{D_t} \sum_{\tau=t-L}^{t} \psi^{(\tau)} \boldsymbol{\theta}^{(\tau)}$$

where $\boldsymbol{\theta}$ refers to either $\boldsymbol{\mu}$ or $\boldsymbol{\Sigma}$, we may now derive a recursive expression for $\boldsymbol{\theta}_t$ like so:

$$\boldsymbol{\theta}_t = \frac{1}{D_t} \left(\left\{ \sum_{\tau=t-L-1}^{t-1} \psi^{(\tau)} \boldsymbol{\theta}^{(\tau)} \right\} + \psi^{(t)} \boldsymbol{\theta}^{(t)} - \psi^{(t-L-1)} \boldsymbol{\theta}^{(t-L-1)} \right)$$

$$= \frac{1}{D_t} \left(D_{t-1} \boldsymbol{\theta}_{t-1} + \psi^{(t)} \boldsymbol{\theta}^{(t)} - \psi^{(t-L-1)} \boldsymbol{\theta}^{(t-L-1)} \right)$$

$$= \frac{1}{D_t} \left(\left\{ \boldsymbol{\theta}_{t-1} \sum_{\tau=t-L-1}^{t-1} \psi^{(\tau)} \right\} + \psi^{(t)} \boldsymbol{\theta}^{(t)} - \psi^{(t-L-1)} \boldsymbol{\theta}^{(t-L-1)} \right)$$

$$= \frac{1}{D_t} \left(\left\{ \boldsymbol{\theta}_{t-1} \sum_{\tau=t-L}^{t} \psi^{(\tau)} \right\} - \psi^{(t)} \boldsymbol{\theta}_{t-1} + \psi^{(t-L-1)} \boldsymbol{\theta}_{t-1} + \psi^{(t)} \boldsymbol{\theta}^{(t)} - \psi^{(t-L-1)} \boldsymbol{\theta}^{(t-L-1)} \right)$$

$$= \frac{1}{D_t} (D_t \boldsymbol{\theta}_{t-1}) + \frac{\psi^{(t)}}{D_t} (\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_{t-1}) + \frac{\psi^{(t-L-1)}}{D_t} (\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^{(t-L-1)})$$

$$= \boldsymbol{\theta}_{t-1} + \frac{\psi^{(t)}}{D_t} (\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_{t-1}) - \frac{\psi^{(t-L-1)}}{D_t} (\boldsymbol{\theta}^{(t-L-1)} - \boldsymbol{\theta}_{t-1})$$

The recursive updates for the corresponding priors $P_t(j)$ are derived similarly.

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision*, pages 469–481, 2004.

[2] T. Ahonen, M. Pietikäinen, A. Hadid, and T. Mäenpää. Face recognition based on the appearance of local regions. In *International Conference on Pattern Recognition*, pages 153–156, 2004.

[3] I. Autio. Using natural class hierarchies in multi-class visual classification. *Pattern Recognition*, 39(7):1290–1299, 2006.

[4] S. Avidan. Ensemble tracking. *Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.

[5] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Mathematics in Science and Engineering. Academic Press Professional, Inc., 1987.

[6] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550, 1994.

[7] A.M. Baumberg and D.C. Hogg. Learning flexible models from image sequences. Technical report, School of Computer Studies, University of Leeds, 1993.

[8] A.M. Baumberg and D.C. Hogg. An efficient method for contour tracking using active shape models. In *Workshop on Motion of Nonrigid and Articulated Objects*, pages 194–199, 1994.

[9] E. Bengtsson. Computerized cell image analysis: Past, present, and future. *Image Analysis*, pages 45–53, 2003.

[10] M. Bertalmío, G. Sapiro, and G. Randall. Morphing active contours. *Pattern Analysis and Machine Intelligence*, 22(7):733–737, 2000.

[11] M. Bichsel. Segmenting simply connected moving objects in a static scene. *Pattern Analysis and Machine Intelligence*, 16:1138–1142, 1994.

[12] C. Bishop. *Neural Networks for Pattern Recognition*. Cambridge University Press, 1995.

[13] A. Blake, M. Isard, and D. Reynard. Learning to track the visual motion of contours. *Artifical Intelligence*, 78:179–212, 1995.

[14] M.B. Blaschko, G. Holness, M.A. Mattar, D. Lisin, P.E. Utgoff, A.R. Hanson, H. Schultz, E.M. Riseman, M.E. Sieracki, W.M. Balch, and B. Tupper. Automatic in situ identification of plankton. *IEEE Workshop on Applications of Computer Vision*, 1:79–86, 2005.

[15] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artifical Intelligence*, 97:245–271, 1997.

[16] K. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical roc curves. *Computer Vision and Image Understanding*, 84(1):77–103, 2001.

[17] D.H. Brainard and B.A. Wandell. Asymmetric color matching: how color appearance depends on the illuminant. *Journal of the Optical Society of America*, 9(9):1433–1448, 1992.

[18] J. Cai and A. Goshtasby. Detecting human faces in color images. *Image and Vision Computing*, 18:63–75, 1999.

[19] E.J. Candes and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine*, 25(2):21–30, 2008.

[20] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[21] R. Caruana and D. Freitag. How useful is relevance? Technical report, In: Relevance, Papers from the 1994 AAAI Fall Symposium, 1994.

[22] A. Cavallaro and T. Ebrahimi. Video object extraction based on adaptive background and statistical change detection. In *SPIE Visual Communications and Image Processing*, volume 4310, pages 465–475, 2000.

[23] Y. Chang, G. Medioni, J. Kang, and I. Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *Pattern Analysis and Machine Intelligence*, 29(9):1627–1641, 2007.

[24] K. Chen, C. Chou, S. Shih, W. Chen, and D. Chen. Feature selection for iris recognition with adaboost. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2:411–414, 2007.

[25] O. Chomat and J.L. Crowley. Probabilistic recognition of activity using local appearance. In *Computer Vision and Pattern Recognition*, pages 104–109, 1999.

[26] R. Clement, M. Dunbabin, and G. Wyeth. Toward robust image detection of crown-of-thorns starfish for autonomous population monitoring. In *Australasian Conference on Robotics and Automation*, 2005.

[27] I. Cohen, A. Garg, and T.S. Huang. Vision-based overhead view person recognition. In *International Conference on Pattern Recognition*, pages 1119–1124, 2000.

[28] S. Cohen. Background estimation as a labeling problem. In *International Conference on Computer Vision*, pages 1034–1041, 2005.

[29] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[30] R.T. Collins and Y. Liu. On-line selection of discriminative tracking features. In *International Conference on Computer Vision*, pages 346–352, 2003.

[31] M. Collobert, R. Feraud, G. Le Tourneur, O.J. Bernier, J.E. Viallet, Y. Mahieux, and D. Collobert. Listen: a system for locating and tracking individual speakers. In *International Conference on Automatic Face and Gesture Recognition*, pages 283–288, 1996.

[32] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition*, pages 142–149, 2000.

[33] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[34] D. Connah and G.D. Finlayson. Using local binary pattern operators for colour constant image indexing. In *European Conference on Color in Graphics, Imaging and Vision*, pages 60–64, 2006.

[35] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[36] T.F. Cootes and C.J. Taylor. 'active shape models - smart snakes. In *British Machine Vision Conference*, pages 266–275, 1992.

[37] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In *British Machine Vision Conference*, pages 9–18, 1992.

[38] M. Corbetta and G.L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Review Neuroscience*, 3(3):201–215, March 2002.

[39] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10:53–66, 1993.

[40] R. Curwen and A. Blake. Dynamic contours: real-time active splines. In *Active Vision*, pages 39–57, 1992.

[41] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[42] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *International Conference on Machine Learning*, pages 74–81, 2001.

[43] J.G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.

[44] J.C. Davidson and S.A. Hutchinson. Recognition of traversable areas for mobile robotic navigation in outdoor environments. In *International Conference on Intelligent Robotics and Systems*, pages 297–304, October 2003.

[45] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annal Review of Neuroscience*, 18:193–222, 1995.

[46] E.D. Dickmanns, R. Behringer, C. Brudigam, D. Dickmanns, F. Thomanek, and V. von Holt. An all-transputer visual autobahn-autopilot/copilot. In *International Conference on Computer Vision*, pages 608–615, 1993.

[47] T.W. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *International Conference on Computer Vision*, pages 315–320, 2001.

[48] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2001.

[49] J. Duncan. Visual search and visual attention. *Attention and Performance XI*, pages 85–105, 1985.

[50] M. Ekinci, F.W.J. Gibbs, and B.T. Thomas. Knowledge-based navigation for autonomous road vehicles. *Turkish Journal of Electrical Engineering and Computer Science*, 8:1–29, 2000.

[51] A.M. Elgammal, D. Harwood, and L.S. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision*, pages 751–767, 2000.

[52] B. S. Everitt and D. J. Hand. *Finite Mixture Distributions*. Chapman and Hall, New York, 1981.

[53] X. Feng, M. Pietikäinen, and A. Hadid. Facial expression recognition with local binary patterns and linear programming. *Pattern Recognition and Image Analysis*, 15:546–548, 2005.

[54] M.D. Fleetwood. *Refining Theoretical Models of Visual Sampling in Supervisory Control Tasks: Examining the Influence of Alarm Frequency, Effort, Value, and Salience*. PhD thesis, Rice University, Houston, TX, 2005.

[55] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[56] J. Fogarty, J. Forlizzi, and S.E. Hudson. Aesthetic information collages: Generating decorative displays that contain information. In *ACM Symposium on User Interface Software and Technology*, pages 141–150, 2001.

[57] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[58] G. Forman, I. Guyon, and A. Elisseeff. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[59] D. A. Forsyth. *Colour Constancy and its Applications in Machine Vision*. PhD thesis, University of Oxford, 1988.

[60] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data-association. *Journal of Oceanic Engineering*, 8(3):173–184, 1983.

[61] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[62] Yoav Freund and Robert E. Schapire. A short introduction to boosting. In *International Joint Conference on Artificial Intelligence*, pages 1401–1406, 1999.

[63] S. Fu and M. Desmarais. Fast markov blanket discovery algorithm via local learning within single pass. In *Canadian Conference on Artificial Intelligence*, pages 96–107, 2008.

[64] X. Gao, T.E. Boult, F. Coetzee, and V. Ramesh. Error analysis of background adaption. *Computer Vision and Pattern Recognition*, 1:503–510, 2000.

[65] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[66] D.M. Gavrila and L.S. Davis. Towards 3d model-based tracking and recognition of human movement: A multi-view approach. In *International Workshop on Automatic Face and Gesture Recognition*, pages 272–277, 1995.

[67] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[68] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[69] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing*, pages 107–113, April 1993.

[70] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition*, pages 260–267, 2006.

[71] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *British Machine Vision Conference*, pages 47–56, 2006.

[72] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *European Conference on Computer Vision*, pages 234–247, 2008.

[73] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C.H. Anderson. Overcomplete steerable pyramid filters and rotation invariance. In *Computer Vision and Pattern Recognition*, pages 222–228, 1994.

[74] I. Guyon. Practical feature selection: from correlation to causality. *Mining Massive Datasets for Security*, pages 27–43, 2008.

[75] I. Guyon, A.R.S.A. Alamdari, G. Dror, and J. Buhmann. Performance prediction challenge. In *International Joint Conference on Neural Networks*, pages 1649–1656, 2006.

[76] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(1):1157–1182, 2003.

[77] I. Guyon, S.R. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *Advances in Neural Information Processing Systems*, pages 545–552, 2004.

[78] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[79] M.A. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.

[80] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[81] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.

[82] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *Advances in Neural Information Processing Systems*, volume 10, 1996.

[83] C. He, T. Ahonen, and M. Pietikäinen. A bayesian local binary pattern texture descriptor. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[84] J.D. Hoffman, J.D. Lee, D.V. McGehee, M. Macias, and A.W. Gellatly. Visual sampling of in-vehicle text messages: Effects of number of lines, page presentation, and message control. *Journal of the Transportation Research Board*, 1937:22–30, 2005.

[85] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *International Conference on Computer Vision*, 2009.

[86] H. Hotelling. Relations between two sets of variables. In *Biometrika 28*, pages 321–377, 1936.

[87] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34:334–352, 2004.

[88] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, 1996.

[89] R. Jaimes and J. Liu. Hotspot components for gesture-based interaction. In *International Conference on Human Computer Interaction*, pages 1062–1066, 2005.

[90] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[91] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *IEEE Workshop on Motion and Video Computing*, pages 22–27, 2002.

[92] H. Jin, Q. Liu, H. Lu, and X. Tong. Face detection using improved lbp under bayesian framework. In *International Conference on Image and Graphics*, pages 306–309, 2004.

[93] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129, 1994.

[94] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[95] J. Kang, I. Cohen, and G. Medioni. Tracking objects from multiple stationary and moving cameras. In *Intelligent Distributed Surveillance Systems*, pages 31–35, 2004.

[96] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, V1(4):321–331, January 1988.

[97] R. Kauth, A. Pentland, and G. Thomas. Blob: An unsupervised clustering approach to spatial preprocessing of mss imagery. In *International Symposium on Remote Sensing of the Environment*, 1977.

[98] V. Kellokumpu, G. Zhao, and Pietikäinen M. Texture based description of movements for activity analysis. In *International Conference on Computer Vision Theory and Applications*, pages 206–213, 2008.

[99] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Human activity recognition using a dynamic texture based method. In *British Machine Vision Conference*, 2008.

[100] R. Kjeldsen and J. Kender. Finding skin in color images. In *International Conference on Automatic Face and Gesture Recognition*, pages 312–317, 1996.

[101] R. Kohavi. *Wrappers for performance enhancement and oblivious decision graphs*. PhD thesis, Stanford University, Stanford, CA, USA, 1996.

[102] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artifical Intelligence*, 97(1-2):273–324, 1997.

[103] D. Koller and M. Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.

[104] H.L. Kundel. Visual sampling and estimates of the location of information on chest films. *Investigative Radiology*, 9(2):87–93, 1974.

[105] T.O. Kvälseth. Human information processing in visual sampling. *Ergonomics*, 21(6):439–54, 1978.

[106] P. Lafferty and F. Ahmed. Texture-based steganalysis: results for color images. In *SPIE Mathematics of Data/Image Coding, Compression, and Encryption, with Applications*, pages 145–151, 2004.

[107] O. Lahdenoja, M. Laiho, and A. Paasio. Reducing the feature vector length in local binary pattern based face recognition. In *International Conference on Image Processing*, pages 914–917, 2005.

[108] P. Langley and S. Sage. Induction of selective bayesian classifiers. In *Conference on Uncertainty in Artificial Intelligence*, pages 399–406, 1994.

[109] U. Leonards, S. Sunaert, P. Van Hecke, and G.A. Orban. Attention mechanisms in visual search - an fmri study. *Journal of Cognitive Neuroscience*, 12:61–75, 2000.

[110] L. Li, W. Huang, Y. Gu, and Q. Tian. Foreground object detection in changing background based on color co-occurrence statistics. In *IEEE Workshop on Applications of Computer Vision*, pages 269–274, 2002.

[111] L. Li, W. Huang, Y. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13:1459–1472, 2004.

[112] L.Y. Li and M.K.H. Leung. Integrating intensity and texture differences for robust change detection. *IEEE Transactions on Image Processing*, 11(2):105–112, February 2002.

[113] S.Z. Li, C. Zhao, M. Ao, and Z. Lei. Learning to fuse 3d+2d based face recognition at both feature and decision levels. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 44–54, 2005.

[114] H. Lian and B. Lu. Multi-view gender classification using local binary patterns and support vector machines. In *International Symposium on Neural Networks*, pages 202–209, 2006.

[115] D.W. Liang, Q.M. Huang, S.Q. Jiang, H.X. Yao, and W. Gao. Mean-shift blob tracking with adaptive feature selection and scale adaptation. In *International Conference on Image Processing*, pages 369–372, 2007.

[116] S. Liao, W. Fan, A.C.S. Chung, and D.Y. Yeung. Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features. In *International Conference on Image Processing*, pages 665–668, 2006.

[117] S. Liao, M.W.K. Law, and A.C.S. Chung. Combining microscopic and macroscopic information for rotation and histogram equalization invariant texture classification. In *Asian Conference on Computer Vision*, pages 100–109, 2006.

[118] S.C. Liao, X.X. Zhu, Z. Lei, L. Zhang, and S.Z. Li. Learning multi-scale block local binary patterns for face recognition. In *International Conference on Biometrics*, pages 828–837, 2007.

[119] R. S. Lin, D. Ross, J. Lim, and M. H. Yang. Adaptive discriminative generative model and its applications. *Advances in Neural Information Processing Systems*, pages 801–808, 2004.

[120] F. López, J.M.V. Valiente, and J.M. Prats. Surface grading using soft colour-texture descriptors. *Lecture Notes in Computer Science*, 3773:13–23, 2005.

[121] D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.

[122] H.C. Lu, C. Wang, and Y.W. Chen. Gaze tracking by binocular vision and lbp features. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[123] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[124] M. Lucassen. *Quantitative Studies of Color Constancy*. PhD thesis, Utrecht University, 1993.

[125] A. Lucieer and A. Stein. Texture-based landform segmentation of LiDAR imagery. *International Journal of Applied Earth Observations and Geoinformation*, 6(3-4):261–270, 2005.

[126] B. Ma, W. Zhang, S. Shan, X. Chen, and W. Gao. Robust head pose estimation using lgbp. In *International Conference on Pattern Recognition*, pages 512–515, 2006.

[127] T. Mäenpää and M. Pietikäinen. Multi-scale binary patterns for texture analysis. In *Scandinavian Conference on Image Analysis*, pages 885–892, 2003.

[128] T. Mäenpää and M. Pietikäinen. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629–1640, 2004.

[129] T. Mäenpää and M. Pietikäinen. Texture analysis with local binary patterns. In *Handbook of Pattern Recognition and Computer Vision*, pages 197–216. World Scientific, 2005.

[130] T. Mäenpää, M. Pietikäinen, and T. Ojala. Texture classification by multipredicate local binary pattern operators. In *International Conference on Pattern Recognition*, pages 3951–3954, 2000.

[131] T. Mäenpää, M. Turtinen, and M. Pietikäinen. Real-time surface inspection by texture. *Real-Time Imaging*, 5(9):289–296, 2003.

[132] P.K. Mallapragada, R. Jin, A.K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *Pattern Analysis and Machine Intelligence*, 31(11):2000–2014, 2008.

[133] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[134] J. Matas, R. Marik, and J. Kittler. On representation and matching of multi-coloured objects. In *International Conference on Computer Vision*, pages 726–732, 1995.

[135] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *Pattern Analysis and Machine Intelligence*, 26(1):810 – 815, June 2004.

[136] J.J. McCann, S.P. McKee, and T.H. Taylor. Quantitative studies in retinex theory. *Vision Research*, 16:445–458, 1976.

[137] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, June 1996.

[138] S.J. McKenna, S. Gong, and Y. Raja. Face recognition in dynamic scenes. In *British Machine Vision Conference*, pages 140–151, 1997.

[139] S.J. McKenna, S. Gong, and Y. Raja. Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 31(12):1883–1892, 1998.

[140] S.J. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive colour mixture models. In *Asian Conference on Computer Vision*, pages 615–622, 1998.

[141] S.J. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3-4):225–231, 1999.

[142] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc., New York, 1988.

[143] H. Mei and T. Kanade. Multiple motion scene reconstruction with uncalibrated cameras. *Pattern Analysis and Machine Intelligence*, 25(7):884–894, 2003.

[144] A.B. Merhy, P. Payeur, and E.M. Petriu. Application of segmented 2d probabilistic occupancy maps for mobile robot sensing and navigation. In *International Instrumentation and Measurement Technology Conference*, pages 2342–2347, 2006.

[145] D. Metaxas and D. Terzopoulos. Shape and non-rigid motion estimation through physics-based synthesis. *Pattern Analysis and Machine Intelligence*, 15:580–591, 1993.

[146] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[147] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modelling and subtraction of dynamic scenes. In *International Conference on Computer Vision*, pages 1305–1312, 2003.

[148] H. T. Nguyen and A. W. Smeulders. Robust tracking using foreground-background texture discrimination. *International Journal of Computer Vision*, 69(3):277–293, 2006.

[149] H.T. Nguyen and A.W. Smeulders. Fast occluded object tracking by a robust appearance filter. *Pattern Analysis and Machine Intelligence*, 26:1099–1104, 2004.

[150] B. Ni and J. Li. A hybrid filter/wrapper gene selection method for microarray classification. In *International Conference on Machine Learning and Cybernetics*, pages 2537–2542, 2004.

[151] I. Novak and Z. Hocenski. Texture feature extraction for a visual inspection of ceramic tiles. In *International Symposium on Industrial Electronics*, pages 1279–1283, 2005.

[152] University of Reading. Pets 2009. http://www.cvg.rdg.ac.uk/PETS2009/, 2009.

[153] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *International Conference on Pattern Recognition*, pages 701–706, 2002.

[154] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.

[155] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. *Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.

[156] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence*, 22:831–843, 1999.

[157] L. Paletta, G. Fritz, and C. Seifert. Reinforcement learning of informative attention patterns for object recognition. In *International Conference on Development and Learning*, pages 188–193, 2005.

[158] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. *Computer Vision and Pattern Recognition*, pages 1034–1040, 2001.

[159] H. Pashler. Attention and visual perception: Analyzing divided attention. In S.M. Kosslyn and D.N. Oshershon, editors, *Visual cognition: Volume 2*, pages 71–100. M.I.T. Press, Boston, MA, 1995.

[160] A.E. Patla, A. Adkin, C. Martin, R. Holden, and S. Prentice. Characteristics of voluntary visual sampling of the environment for safe locomotion over different terrains. *Experimental Brain Research*, 112(3):513–522, December 1996.

[161] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *Pattern Analysis and Machine Intelligence*, 19:677–695, 1997.

[162] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

[163] A. Pentland. Classification by clustering. In *IEEE Symposium on Machine Processing and Remotely Sensed Data*, 1976.

[164] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *Pattern Analysis and Machine Intelligence*, 13(7):730–742, 1991.

[165] M. Pietikäinen, Z. Xu, and T. Ojala. Rotation-invariant texture classification using feature distributions. In *Scandinavian Conference on Image Analysis*, pages 103–110., 1997.

[166] M. K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599, 1999.

[167] R. Plaenkers and P. Fua. Model-based silhouette extraction for accurate people tracking. In *European Conference on Computer Vision*, pages 325–339, 2002.

[168] C. E. Priebe. Adaptive mixtures. *Journal of the American Statistics Association*, 89(427):796–806, 1994.

[169] C. E. Priebe and D. J. Marchette. Adaptive mixtures: Recursive nonparametric pattern recognition. *Pattern Recognition*, 24(12):1197–1209, 1991.

[170] C. E. Priebe and D. J. Marchette. Adaptive mixture density estimation. *Pattern Recognition*, 26(5):771–785, 1993.

[171] O. Pujol and P. Radeva. Supervised texture classification for intravascular tissue characterization, 2005.

[172] X. Qing, Y. Jie, and D. Siyi. Texture segmentation using lbp embedded region competition. *Electronic Letters on Computer Vision and Image Analysis*, 5:41–47, 2005.

[173] Y. Raja and S. Gong. Robust tracking by adaptive multi-feature association. Submitted to International Journal of Computer Vision, 2010.

[174] Y. Raja and S. Gong. Sparse multiscale local binary patterns. In *British Machine Vision Conference*, pages 799–808, 2006.

[175] Y. Raja, S.J. McKenna, and S. Gong. Colour model selection and adaption in dynamic scenes. In *European Conference on Computer Vision*, pages 460–474, 1998.

[176] Y. Raja, S.J. McKenna, and S. Gong. Segmentation and tracking using colour mixture models. In *Asian Conference on Computer Vision*, pages 607–614, 1998.

[177] Y. Raja, S.J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. In *International Conference on Automatic Face and Gesture Recognition*, pages 228–233, 1998.

[178] Y. Raja, S.J. McKenna, and S. Gong. Using colour for robust object tracking and segmentation. In *Noblesse Workshop on Non-linear Model Based Image Analysis*, pages 199–204, 1998.

[179] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.

[180] J. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European Conference on Computer Vision*, pages 35–46, 1994.

[181] G.C. Ricker and J.R. Williams. Adaptive tracking filter for maneuvering targets. In *Aerospace and Electronic Systems*, volume 14, pages 185–193, 1978.

[182] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. In *European Conference on Computer Vision*, pages 336–350, 2000.

[183] Y. Rodriguez and S. Marcel. Face authentication using adapted local binary pattern histograms. In *European Conference on Computer Vision*, pages 321–332, 2006.

[184] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):94–115, 1994.

[185] D.M. Russell and S.G. Gong. Minimum cuts of a time-varying background. In *British Machine Vision Conference*, pages 809–818, 2006.

[186] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[187] M.A. Savelonas, D.K. Iakovidis, and D. Maroulis. Lbp-guided active contours. *Pattern Recognition Letters*, 29(9):1404–1415, 2008.

[188] D. Saxe and R. Foulds. Toward robust skin identification in video images. In *International Conference on Automatic Face and Gesture Recognition*, pages 379–384, 1996.

[189] B. Schiele and A. Waibel. Gaze tracking based on face-color. In *International Workshop on Automatic Face and Gesture Recognition*, pages 344–349, 1995.

[190] R.G. Sea. An efficient suboptimal decision procedure for associating sensor data with stored tracks in real-time surveillance systems. In *Conference on Decision and Control*, pages 33–37, 1971.

[191] M. Sebban and R. Nock. A hybrid filter/wrapper approach of feature selection using information theory. *Pattern Recognition*, 35(4):835–846, April 2002.

[192] J.W. Senders, A.B. Kristofferson, W.H. Levison, C.W. Dietrich, and J.L. Ward. The attentional demand of automobile driving. *Highway Research Record*, 195:15–32, 1967.

[193] C. Shan, S. Gong, and P. McOwan. Conditional mutual information based boosting for facial expression recognition. In *British Machine Vision Conference*, pages 399–408, 2005.

[194] C. Shan, S. Gong, and P. McOwan. Robust facial expression recognition using local binary patterns. In *International Conference on Image Processing*, pages 370–373, 2005.

[195] C. Shan, S. Gong, and P.W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.

[196] C.F. Shan and T. Gritti. Learning discriminative lbp-histogram bins for facial expression recognition. In *British Machine Vision Conference*, 2008.

[197] L. Shen, L. Bai, D. Bardsley, and Y. Wang. Gabor feature selection for face recognition using improved adaboost learning. *Advances in Biometric Person Authentication*, pages 39–49, 2005.

[198] P. Silapachote, D.R. Karuppiah, and A.R. Hanson. Feature selection using adaboost for face expression recognition. In *International Conference on Visualization, Imaging, and Image Processing*, pages 1173–1185, 2005.

[199] R.A. Singer and J.J. Stein. An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems. In *Conference on Decision and Control*, pages 171–175, 1971.

[200] R.W. Sittler. An optimal data association problem in surveillance theory. *IEEE Transactions on Military Electronics*, pages 125–139, 1964.

[201] W. Skarbek and A. Koschan. Colour image segmentation - a survey. Technical report, Tech. Univ. of Berlin, 1994.

[202] L. Snidaro, I. Visentini, and G. Foresti. Dynamic models for people detection and tracking. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 29–35, 2008.

[203] K. Sobottka and I. Pitas. Segmentation and tracking of faces in color images. In *International Conference on Automatic Face and Gesture Recognition*, pages 236–241, 1996.

[204] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *European Conference on Computer Vision*, pages 642–655, 2008.

[205] M. Soriano, S. Marcos, M. Quibilan, P. Alino, and C. Saloma. Image classification of coral reef components from underwater color video. In *IEEE OCEANS 2001*, pages 1008–1013, 2001.

[206] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition*, 2:2246, 1999.

[207] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[208] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Buhmann. Topology free hidden markov models: Application to background modeling. In *International Conference on Computer Vision*, pages 294–301, 2001.

[209] R. Stolkin, I. Florescu, and G. Kamberov. An adaptive background model for camshift tracking with a moving camera. In *Advances in Pattern Recognition*, pages 147–151, 2007.

[210] N. Sun, W. Zheng, C. Sun, C. Zou, and L. Zhao. Gender classification based on boosting local binary pattern. In *International Symposium on Neural Networks*, pages 194–201, 2006.

[211] Z. Sun, T. Tan, and X. Qiu. Graph matching iris image blocks with local binary pattern. In *International Conference on Biometrics*, pages 366–372, 2006.

[212] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[213] K. Taeho and J. Kang-Hyun. Detection of moving object using remained background under moving camera. *International Journal of Information Acquisition*, 4(3):227–236, 2007.

[214] V. Takala and M. Pietikainen. Multi-object tracking using color, texture and motion. In *International Workshop on Visual Surveillance*, pages 1–7, 2007.

[215] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 168–182, 2007.

[216] X. Tan and B. Triggs. Fusing gabor and lbp feature sets for kernel-based face recognition. In *Analysis and Modelling of Faces and Gestures*, volume 4778, pages 235–249, oct 2007.

[217] F.B. Tek, A.G. Dempster, and I. Kale. Parasite detection and identification for automated thin blood film malaria diagnosis. *Computer Vision and Image Understanding*, pages 21–32, August 2009.

[218] J. Theeuwes. Cross-dimensional perceptual selectivity. *Perception and Psychophysics*, 50:184–193, 1991.

[219] J. Theeuwes. Perceptual selectivity for color and form. *Perception and Psychophysics*, 51(6):599–606, 1992.

[220] B.T. Thomas, E.L. Dagless, D.J. Milford, and A.D. Morgan. Real-time vision-guided navigation. *Engineering Application of Artificial Intelligence*, 4:287–300, 1991.

[221] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley, New York, 1985.

[222] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *International Conference on Computer Vision*, pages 255–261, 1999.

[223] H. G. C. Traven. A neural network approach to statistical pattern classification by "semi-parametric" estimation of probability density functions. *IEEE Transactions on Neural Networks*, 2(3):366–378, 1991.

[224] I. Tsamardinos and C. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *International Workshop on Artificial Intelligence and Statistics*, 2003.

[225] I. Tsamardinos, C. Aliferis, A. Statnikov, and E. Statnikov. Algorithms for large scale markov blanket discovery. In *International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, pages 376–380, 2003.

[226] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[227] C. Urdiales, T.J. Rubio, and F. Sandoval. Our world from above: texture based segmentation for landscape analysis, 2004.

[228] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[229] X. Wang, H. Gong, H. Zhang, B. Li, and Z. Zhuang. Palmprint identification using boosting local binary pattern. In *International Conference on Pattern Recognition*, pages 503–506, 2006.

[230] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[231] C.D. Wickens, J. Helleberg, J. Goh, X. Xu, and W.J. Horrey. Pilot task management: Testing an attentional expected value model of visual scanning. Technical Report ARL-01-14/NASA-01-7, NASA Ames Research Center, 2001.

[232] W.G. Wierwille. Visual and manual demands of in-car controls and displays. In B. Peacock and W. Karwowski, editors, *Automotive Ergonomics*, pages 299–320. Taylor and Francis, New York, 1993.

[233] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *Pattern Analysis and Machine Intelligence*, 22(8):774–780, 2000.

[234] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder:real-time tracking of the human body. *Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[235] H. Wu, Q. Chen, and M. Yachida. An application of fuzzy theory: face detection. In *International Workshop on Automatic Face and Gesture Recognition*, pages 314–319, June 1995.

[236] J. Wu, S.C. Brubaker, M.D. Mullin, and J.M. Rehg. Fast asymmetric learning for cascade face detection. *Pattern Analysis and Machine Intelligence*, 30(3):369–382, 2008.

[237] X. Wu and B. Bhanu. Gabor wavelets for 3-d object recognition. *International Conference on Computer Vision*, pages 537–542, 1995.

[238] S. Xie, S. Shan, X. Chen, and W. Gao. V-lgbp: Volume based local gabor binary patterns for face representation and recognition. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[239] P. Xu, R. Ji, H. Yao, X. Sun, T. Liu, and X. Liu. Text particles multi-band fusion for robust text detection. In *International conference on Image Analysis and Recognition*, pages 587–596, 2008.

[240] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning*, pages 412–420, 1997.

[241] S. Yaramakala. Fast markov blanket discovery. Master's thesis, Iowa State University, 2004.

[242] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):1–45, 2006.

[243] A. Yilmaz, X. Li, and M. Shah. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *Pattern Analysis and Machine Intelligence*, 26:1531–1536, 2004.

[244] Q. Yu, T.B. Dinh, and G.G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *European Conference on Computer Vision*, pages 678–691, 2008.

[245] B.D. Zarit, B.J. Super, and F.K.H. Quek. Comparison of five color models in skin pixel classification. *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 58–63, 1999.

[246] H. Zhang, W. Gao, X. Chen, and D. Zhao. Learning informative features for spatial histogram-based object detection. In *International Joint Conference on Neural Networks*, pages 1806–1811, 2005.

[247] H. Zhang and D. Zhao. Spatial histogram features for face detection in color images. In *Pacific Rim Conference on Multimedia*, pages 377–384, 2004.

[248] L. Zhang, R. Chu, S. Xiang, S. Liao, and S.Z. Li. Face detection based on multi-block lbp representation. In *International Conference on Biometrics*, pages 11–18, 2007.

[249] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In *International Conference on Computer Vision*, pages 786–791, 2005.

[250] W. Zhang, S. Shan, H. Zhang, W. Gao, and X. Chen. Multi-resolution histograms of local variation patterns (mhlvp) for robust face recognition. In *International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 937–944, 2005.

[251] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[252] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Computer Vision and Pattern Recognition*, pages 819–826, 2004.

[253] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. *International Conference on Computer Vision*, 1:44–50, 2003.

[254] H. Zhou, R. Wang, and C. Wang. A novel extended local-binary-pattern operator for texture analysis. *Information Sciences*, 178(22):4314–4325, 2008.

[255] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang. Hmm-based acoustic event detection with adaboost feature selection. In *Multimodal Technologies for Perception of Humans*, pages 345–353, 2008.